Solution Building Report on Text De-toxification
Practical Machine Learning and Deep Learning - Assignment 1

Jayveersinh Raj,
BS20-DS01,
j.raj@innoplis.university

Abstract

Text detoxification, also known as text cleaning or profanity filtering, is a crucial task in natural language processing aimed at removing offensive, harmful, or inappropriate content from text data. In an era where online communication is prevalent, ensuring clean and respectful language is essential for maintaining a positive digital environment. Text Detoxification task is a process of transforming the text with toxic style into the text with the same meaning but with neutral style. Text detoxification involves identifying and filtering out profane words, hate speech, or other forms of inappropriate language to create a sanitized version of the text. This process finds applications in social media platforms, customer service, content moderation, and various online communities, fostering safer and more inclusive online spaces.

1. Data Exploration and Preparation

1.1. Description

The dataset is a subset of the ParaNMT corpus (50M sentence pairs). The dataset has the following columns with corresponding description:

1. **reference**: First item from the pair
2. **ref_tox**: toxicity level of reference text
3. **translation**: Second item from the pair - paraphrazed version of the reference
4. **trn_tox**: toxicity level of translation text
5. **similarity**: cosine similarity of the texts
6. **lenght_diff**: relative length difference between texts

1.2. Insights from exploratory data analysis

The data has **577,777** data points. Moreover, it was discovered that in some data samples, **translation** did not necessarily contained the detoxified text, but reference was detoxified and corresponding translation had toxicity. One example of such can be seen below:

reference	translation	similarity	lenght_diff	ref_tox	trn_tox
You didn't know that Estelle had stolen some f...	you didn't know that Estelle stole your fish f...	0.870322	0.030769	0.000121	0.949143
It'll suck the life out of you!	you'd be sucked out of your life!	0.722897	0.058824	0.996124	0.215794

It can be seen that in the first example above translation toxicity (**trn_tox**) is more than 90 while reference toxicity (**ref_tox**) is less than 0.1, while on the contrary second example tranlstion toxicity is approximately around 0.2 while reference toxicity is more than 90.

1.3. Training data preparation

Based on the findings from above section, for the training data preparation a new empty pandas DataFrame was created to store the sentence pairs. By iterating over the dataframe with raw data, the **ref_tox** and **trn_tox** were checked, the one with higher toxicity was considered to be a toxic sentence, and hence stored in the newly created empty DataFrame as a column **toxic_texts**, while the one with corresponding less toxicity was stored in the empty DataFrame was **detoxified**. Moreover, for checking the values if the toxicity only texts with greater than 50% toxicity and corresonding detoxification with less than 50% toxicity level were considered. All the data samples had at least translation or reference with greater than 90% toxicity and corresponding other with less than 50% detoxification. Hence, all the samples were training ready. For this data creation **100k** samples were selected from a slice between **400k-500k**. The following choice of sample size is discussed in the models and solution sections. However, for choosing from this particular slice is random.

1.4. Test data preparation

The test data contained **1000** samples to make evaluations. The test size was kept smaller intentionally, the sections further of solutions would explain this choice given the limited computational capabilities. For this data creation **100k** samples were selected were selected from a slice between **400k-500k**, and out of which 0.01% that is 1000 samples were created using a split with **seed 42**.

2. Baseline solution

2.1. Dataset details

A dictionary based baseline was created using the dictionary data from Kaggle https://www.kaggle.com/datasets/nicapotato/bad-bad-words. The dataset contained **1616** bad words.

2.1.1. *Implementation details*

Based on the aforementioned dictionary, if there were bad words in the text, they were just removed using regex while converting all the words to lower case.

2.2. *Evaluation results*

The following evaluations were considered with the corresponding reasons:

- ROUGE (Recall-Oriented Understudy for Gisting Evaluation) scores: For measuring overlap of n-grams.
- BERT embedding similarity: For measuring semantic similarity.
- A toxicity classifier pre-trained model's predictions: To check how many detoxified sentences still contained toxicity.

2.2.1. *ROUGE Metrics*

Average ROUGE (Recall-Oriented Understudy for Gisting Evaluation) scores for the solution on the test set are as follows:

- Average ROUGE-1: 0.542
- Average ROUGE-2: 0.289
- Average ROUGE-L: 0.523

2.2.2. *BERT embedding similarity*

The test set was tokenized and the mean cosine similarity was calculated to be **0.92**.

2.2.3. *Toxicity classifier*

A toxicity classifier (based on XLM-R multilingual embedding and XLM-R classifier head) with 278M parameters (https://huggingface.co/Jayveersinh-Raj/PolyGuard) was used to find the total number of toxic sentences in the ground truth column of the test set which was found to be **355**, and as expected the baseline corrected also has exactly the same number of toxic sentences because the solution simply just removes the toxic words.

2.3. *Problems*

Although the numbers seems promising, but it is very important to remember that just removing the toxic words or replacing them with a corresponding less toxic word not necessarily solve the problem. One of the example is an example from the solution which reflects the problem.

```
("I don't give a shit.", "i don't give a  .")
```

The above sentence lost its meaning, and no word was replaced or the sentence was paraphrased, and in fact the sentence became wrong and incomplete. To address this problem a hypothesis that since the dataset is large, and entire phrasing of the sentence is required, models with pre-existing knowledge base on the language might be a promising solution, hence, a sequence to sequence language model (seq2seqLM) described below was trained.

3. Hypothesis 1: Sequence to Sequence language model (Seq2SeqLM) fine-tuning

3.1. *Training and test data preparation*

As mentioned in the training data subsection of data preparation **100k** samples were selected, but only **90k** were actually used to train the model, the rest were kept for testing, however, it was discovered that it would be difficult to evaluate such big data especially due to computational complexity. This **1k** test set is consistent as discussed in the above sections. The train set was prefixed with **dtox** to guide the model for task it is supposed to do, but since it is being trained for a singular task that is detoxification of text it is optional, and does not influence the results.

3.2. *Model architecture*

t5-small model which is a text to text small language model which supports English, French, Romanian, German, was fine tuned for detoxification task. Input was toxic sentences and the output were the detoxified sentences. The following are the model details and hyper-parameters:

- Parameters: 60.5M
- Epochs: 3
- Half-precision floating-point format (fp16)
- Learning rate: 2e-5
- Weight decay: 0.01

3.3. *Evaluation*

Like the baseline, this model was also evaluated on the same test set of 1000 samples, with same strategies. The following are the results:

- ROUGE1, ROUGE2, ROUGE-L respectively: 0.575, 0.332, 0.554
- BERT embedding similarity: 0.94
- Toxicity classification toxic label count: 602 (ground truth toxicity classification toxic label count: 355)

3.4. *Problems*

This solution of sequence to sequence language model (seq2seqLM) for text to text generation is better than the baseline in all aspects, not just the metrics, but it also solves the problem of incomplete sentences, and rephrases the texts. However, from the toxicity model as it can be seen that **602** rephrased sentences are still toxic while in the ground truth of the corresponding input toxic sentences in the test dataset, there are only **355** sentences with toxic labels (out of 1000, annotated by humans). The above results mean that there are still 255 labels that are rephrased as toxic, hence the model makes more mistakes than a human as human error on 1000 samples were nearly 35 % while the model makes nearly 60 % hence, there is still a room for improvement.

4. Hypothesis 2: Prompt-tuning large language model using Quantized Low Rank Adaptation (QLoRA)

Based on Hypothesis 1's promising results, its quite evident that language models because of its huge knowledge base on the language creates promising results. Therefore, building on hypothesis 1, hypothesis 2 takes 1 step advanced by prompt tuning a large language model by freezing its most of the layers, and only training on few parameters unlike hypothesis 1 (seq2seqLM) where the entire pre-trained model is usually trained in fine-tuning, including all its layers and parameters. Moreover, due to computational limitations, model was trained on 8 bit quantization, and using low rank adaptation. This is the final solution, and is discussed in detail in the Final Solution Report.