# Assignment 1 - Text De-toxification - Report 2

Sergey Golubev, B20-AI-01 student

s.golubev@innopolis.university

Solution link

# EDA

*notebooks/1.0-EDA.ipynb*
*notebooks/1.1-data-preparation.ipynb*

After looking at the proposed ParaNMT subset, I explored that **the fields `reference` and `translation` are often mixed in terms of toxicity level**.

In my solution, I introduce the fields `source` and `target` and distribute the samples so that `toxicity(source) > toxicity(target)`.

`source` and `target` fields will be a better choice for training and fine-tuning the models.

### Additional Dataset

Recently I discovered a paradetox dataset for English de-toxification task. It is a perfect fit for this assignment. Unfortunately, I didn’t managed to use it due to lack of time. Merging it with the given dataset and fine-tuning the models on the result could be a great step to increase the metrics.

# Toxicity Metric

*notebooks/2.0-toxicity-metric.ipynb*

In my solution, I have a toxicity metric that allows to:

1. choose the best output option if a model provides several;
2. evaluate the models.

To evaluate toxicity, I use the toxicity classifier trained by Skolkovo Intitute (link). The model itself is a RoBERTa fine-tuned on toxicity binary classification task.

I retrieve **both toxicity label (True/False) and toxicity metric (float number)** for a given text sample.

- The toxicity metric is retrieved as the model output layer logits (the smaller the value, the more toxic the text).
- The toxicity label is assigned as it is done inside the model, i.e. according to the logits value (positive => non-toxic, negative => toxic).

# Model 1: Fine-tuned T5 for paraphrasing

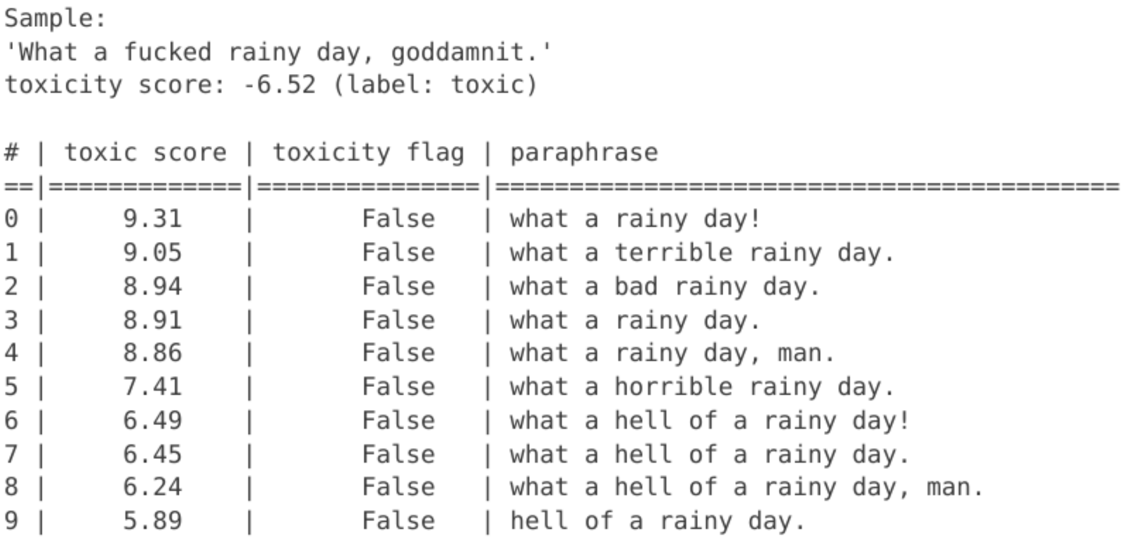
*3.0-t5-finetuning.ipynb*
*3.1-t5-inference.ipynb*

I decided to fine-tune the T5 model to paraphrase examples from the dataset. I use the fine-tuned model to generate several paraphrased options for an input, and then choose the one with minimum toxicity. The toxicity metric is described in the First Report.

I reference the "detox” repository as a source of inspiration for this assignment. For example, I took the T5 model fine-tuned by Skolkovo Institute for text paraphrasing (link), and then fine-tuned it on the processed dataset.

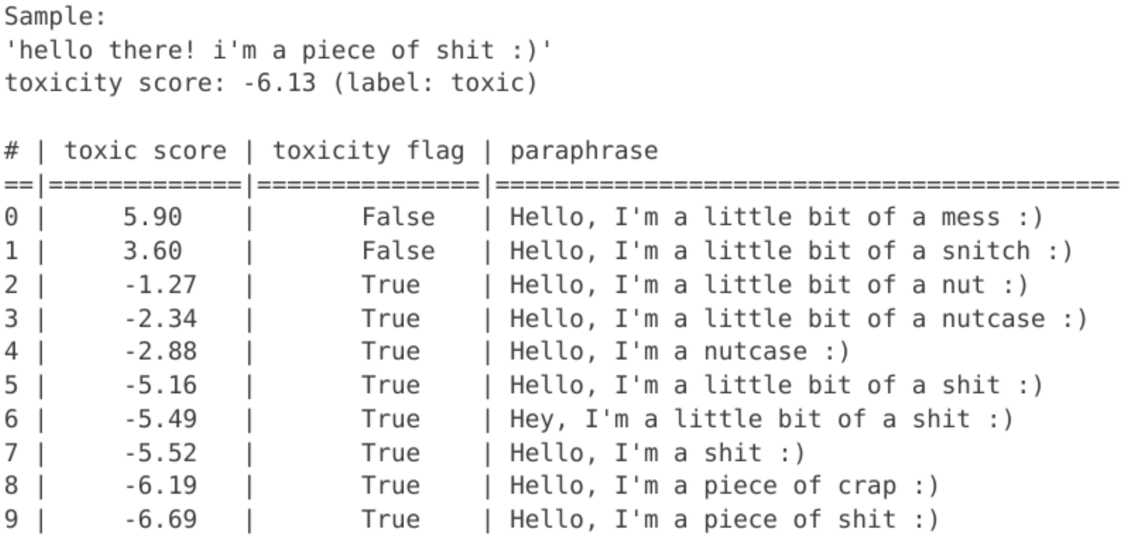
- The T5 model by the link above is fine-tuned for paraphrasing on PAWS, MSRP, Opinosis, and ParaNMT datasets which is quite a big corpus.
- I fine-tuned it further using the ParaNMT toxicity subset that is given for this assignment (and additionally processed in the notebooks `1.x`).

Inference examples:



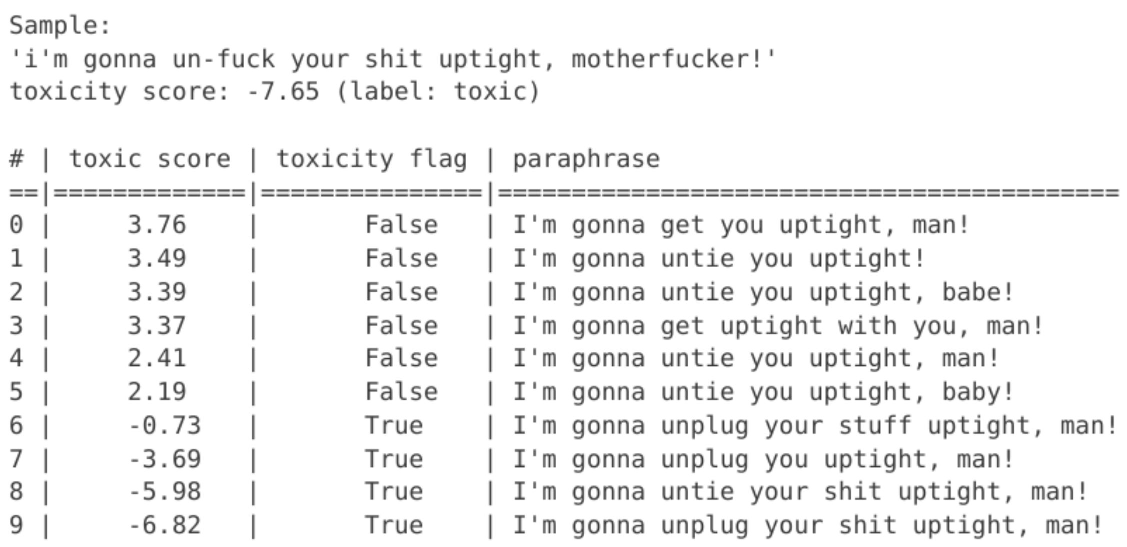
```
Sample:
'What a fucked rainy day, goddamnit.'
toxicity score: -6.52 (label: toxic)

# | toxic score | toxicity flag | paraphrase
==|=============|===============|==========================================
0 |     9.31    |       False   | what a rainy day!
1 |     9.05    |       False   | what a terrible rainy day.
2 |     8.94    |       False   | what a bad rainy day.
3 |     8.91    |       False   | what a rainy day.
4 |     8.86    |       False   | what a rainy day, man.
5 |     7.41    |       False   | what a horrible rainy day.
6 |     6.49    |       False   | what a hell of a rainy day!
7 |     6.45    |       False   | what a hell of a rainy day.
8 |     6.24    |       False   | what a hell of a rainy day, man.
9 |     5.89    |       False   | hell of a rainy day.
```



```
Sample:
'hello there! i'm a piece of shit :)'
toxicity score: -6.13 (label: toxic)

# | toxic score | toxicity flag | paraphrase
==|=============|===============|==========================================
0 |     5.90    |       False   | Hello, I'm a little bit of a mess :)
1 |     3.60    |       False   | Hello, I'm a little bit of a snitch :)
2 |     -1.27   |       True    | Hello, I'm a little bit of a nut :)
3 |     -2.34   |       True    | Hello, I'm a little bit of a nutcase :)
4 |     -2.88   |       True    | Hello, I'm a nutcase :)
5 |     -5.16   |       True    | Hello, I'm a little bit of a shit :)
6 |     -5.49   |       True    | Hey, I'm a little bit of a shit :)
7 |     -5.52   |       True    | Hello, I'm a shit :)
8 |     -6.19   |       True    | Hello, I'm a piece of crap :)
9 |     -6.69   |       True    | Hello, I'm a piece of shit :)
```



```
Sample:
'i'm gonna un-fuck your shit uptight, motherfucker!'
toxicity score: -7.65 (label: toxic)

# | toxic score | toxicity flag | paraphrase
==|=============|===============|==========================================
0 |     3.76    |      False    | I'm gonna get you uptight, man!
1 |     3.49    |      False    | I'm gonna untie you uptight!
2 |     3.39    |      False    | I'm gonna untie you uptight, babe!
3 |     3.37    |      False    | I'm gonna get uptight with you, man!
4 |     2.41    |      False    | I'm gonna untie you uptight, man!
5 |     2.19    |      False    | I'm gonna untie you uptight, baby!
6 |     -0.73   |      True     | I'm gonna unplug your stuff uptight, man!
7 |     -3.69   |      True     | I'm gonna unplug you uptight, man!
8 |     -5.98   |      True     | I'm gonna untie your shit uptight, man!
9 |     -6.82   |      True     | I'm gonna unplug your shit uptight, man!
```

# Model 2: CondBERT Inference

*4.0-condbert-inference.ipynb*

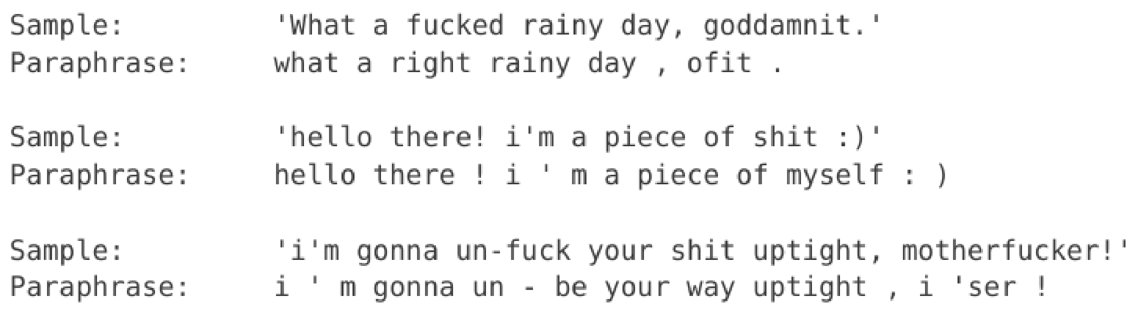
As the second solution, I decided to perform an inference of the **CondBERT model** for text de-toxification proposed in the paper (the code from `src/models/condbert/condBERT` is provided by detox/condBERT).

This model is essentially a zero-shot BERT applied to tokens replacement task. We firstly replace the toxic words with the `[MASK]` token, and then the BERT is inferred to replace the masks with words that fit the surrounding context.

This model performs much worse compared to T5. The reasons are the following:

- The initial sentence intent may not be preserved, especially in short sentences, where each token may carry significant semantic weight. In this case, masking of only one token could result in sense missing, or ambiguity.
- Although the toxicity is removed, we often see some artifacts instead of removed words.

Inference examples:



```
Sample:        'What a fucked rainy day, goddamnit.'
Paraphrase:    what a right rainy day , ofit .

Sample:        'hello there! i'm a piece of shit :)'
Paraphrase:    hello there ! i ' m a piece of myself : )

Sample:        'i'm gonna un-fuck your shit uptight, motherfucker!'
Paraphrase:    i ' m gonna un - be your way uptight , i 'ser !
```

# Data & Models Downloading

- To download and prepare the dataset and the T5 fine-tuned model weights, simply run the `src/start.py` file from your IDE, or with a command:

  ```
  python src/start.py
  ```

- The files `models/condbert/condbert_inference.py` and `models/t5_parphrase/t5_inference.py` contain methods `detoxify()` **used for inference**.
- The file `models/t5_parphrase/t5_train.py` contains method `train()` **to fine-tune the T5 model**.

# Results

I've compared the fine-tuned T5 and the CondBERT models on the validation set. Two metrics are evaluated:

- **toxicity_percentage -** the percentage of the model outputs classified as 'toxic'
- **avg_toxicity** - the toxicity level averaged by the model outputs

The results are as follows:

| **column** | **toxicity_percentage** | **avg_toxicity** |
|---|---|---|
| source | 0.830 | 3.414778 |
| target | 0.087 | 5.573105 |
| t5 | 0.103 | 6.222356 |
| condbert | 0.066 | 6.672241 |

As you can see, the condbert model is the best one according to the introduced metrics :)

- However, **we have no fluency metric** which is also an important one when talking about text seq2seq task. Condbert's generations often look clumsy and distort the initial message meaning. So, if I have managed to implement some fluency metric, the condbert would certainly perform poorly with it.
- We also see that t5 generations on average has less toxicity compared to the `target` column (6.22 > 5.57). However, t5's toxicity percentage is slightly greater than the target's one.

I suppose, the most valuable result of this work is **a powerful fine-tuned T5 paraphraser model**. It is capable of accurate English de-toxification while preserving the initial message intent.