Introduction

The main task of this assignment is text detoxification. Text detoxification is a process of transforming the text with toxic style into the text with the same meaning but with neutral style.

Data analysis

	Unnamed: 0	reference	translation	similarity	lenght_diff	ref_tox	trn_tox
0	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	1	Now you're getting nasty.	you're becoming disgusting.	0.749687	0.071429	0.065473	0.999039
2	2	Well, we could spare your life, for one.	well, we can spare your life.	0.919051	0.268293	0.213313	0.985068
3	3	Ah! Monkey, you've got to snap out of it.	monkey, you have to wake up.	0.664333	0.309524	0.053362	0.994215
4	4	I've got orders to put her down.	I have orders to kill her.	0.726639	0.181818	0.009402	0.999348

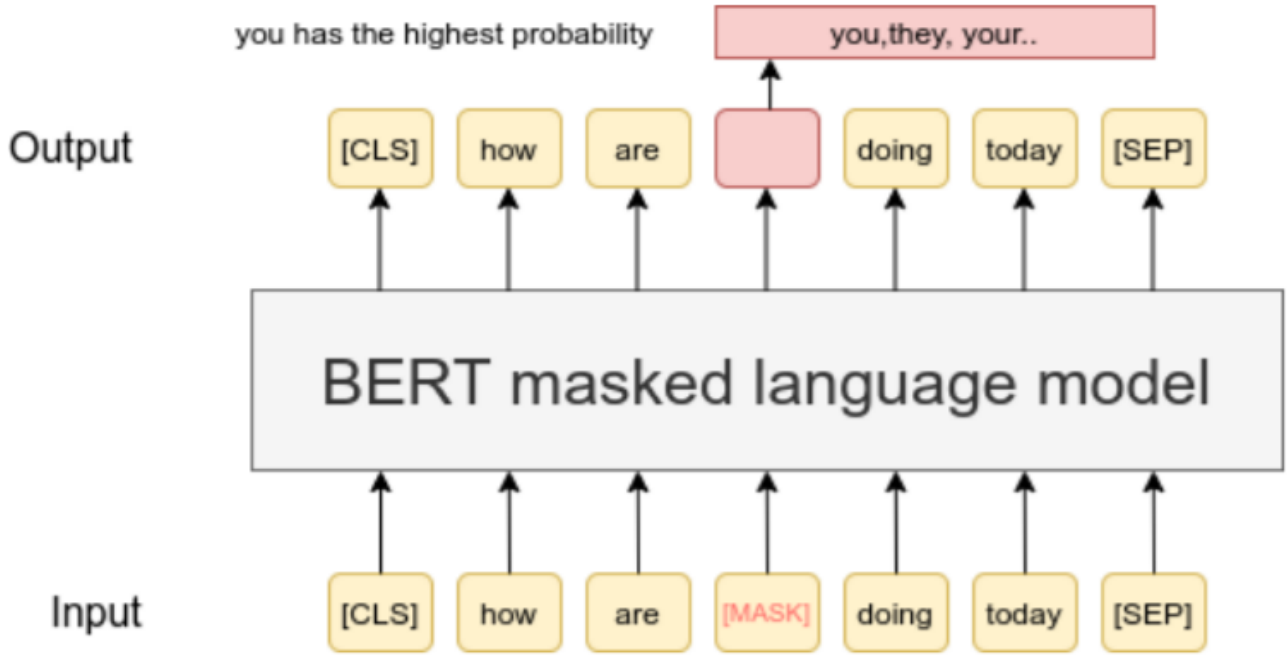
The dataset consists of 6 columns and 577779 rows.
Columns:

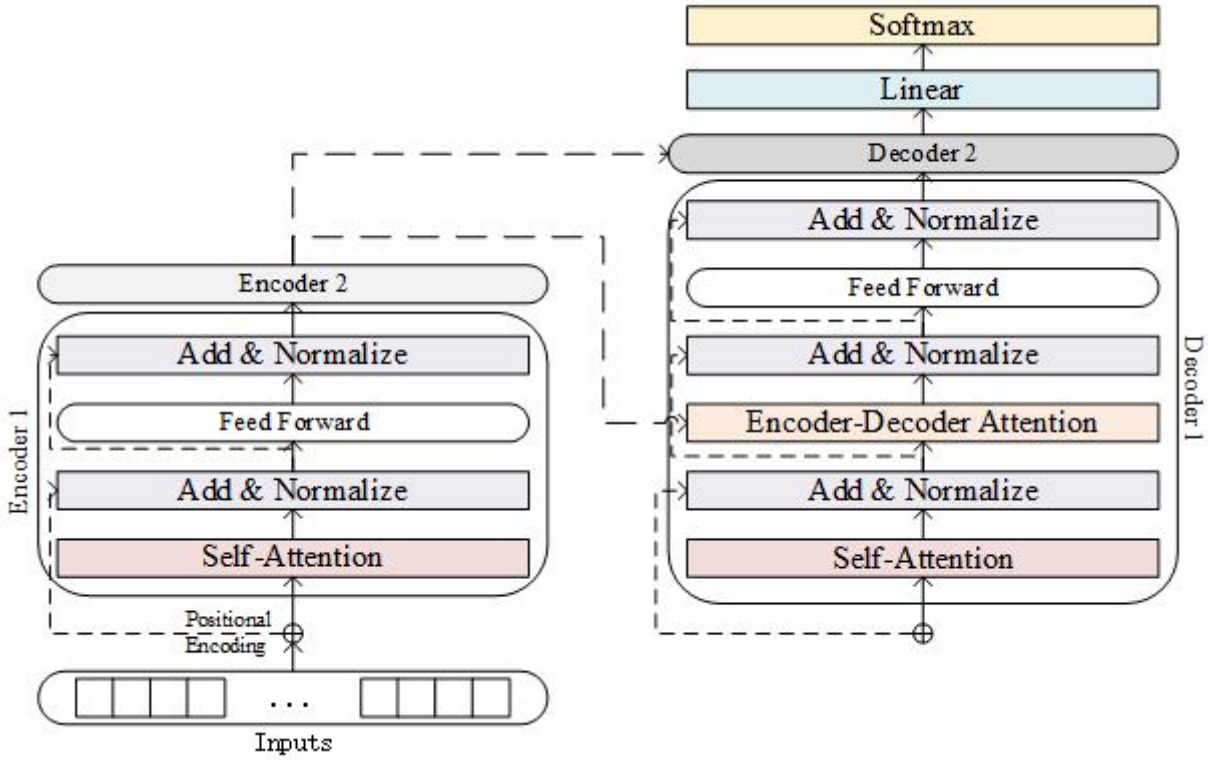
- reference - initial sentence
- translation - translated sentence
- similarity - cosine of the texts
- length_diff - relative length difference between texts
- ref_tox - toxicity score of initial sentence
- trn_tox - toxicity score of translated sentence
  The main observation of this part is that reference sentence can be less toxic than translation sentence. It means that we have at least 2 options:
- We can just remove all such cases (Where reference sentence is more toxic than translated)
- We can rearrange translated sentence and reference sentence
  I found some rows with inappropriate data. I mean not really toxic data for reference and not really non-toxic data for translation. I want to delete such columns because it negatively affects on the training stage.

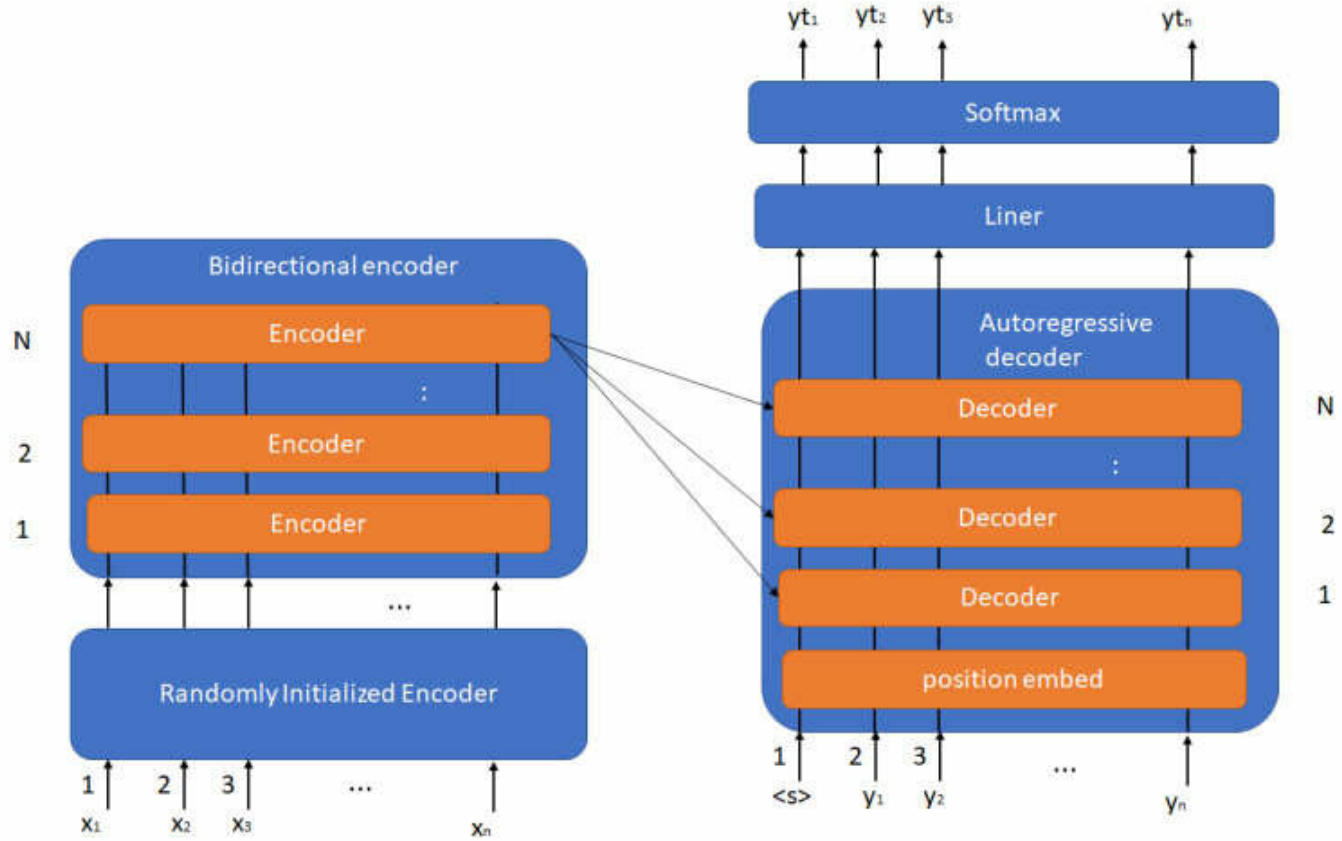
Model Specification

I tested various models: BERT, Bart, T5, dictionary based method as baseline. I have explored a range of models, including BERT, Bart, T5, and a dictionary-based method as a baseline. This comprehensive testing approach allows for a thorough evaluation of various model architectures and techniques, each with its unique strengths and weaknesses. BERT, Bart, and T5, being transformer-based models, offer powerful capabilities for understanding and generating natural language text. On the other hand, the dictionary-based method serves as a useful reference point, especially in evaluating the performance of the more advanced models.

By systematically testing and comparing these models, I determine which one aligns best with the objectives and requirements of text-detoxification.







Training process

I used various models, such as BERT, Bart, T5, and even baseline dictionary-based methods. During the training process, these models are exposed to vast amounts of data, enabling them to learn and extract patterns, relationships, and context from the information provided. The

training procedure involves adjusting the model's parameters, optimizing its performance on specific tasks, and fine-tuning it to the problem at hand. The choice of model, the quality and quantity of training data, and the training methodology all impact the model's effectiveness. By testing a range of models, including state-of-the-art transformer models like BERT, Bart, and T5 I found the best one.
Due to limited computational resources, I had to work with a subset of the initial dataset. Consequently, my training dataset comprised 16,000 rows, while the test and validation datasets contained 2,000 rows each. In my modeling process, I trained the T5 model for 10 epochs and the BERT model for 13 epochs. However, I encountered overfitting issues with the T5 pretrained and Bart models during their initial training, which prompted me to fine-tune them for only 2 epochs. In the first attempt, I trained these models for 5 epochs, but overfitting became evident, leading me to make the decision to reduce the number of training epochs.

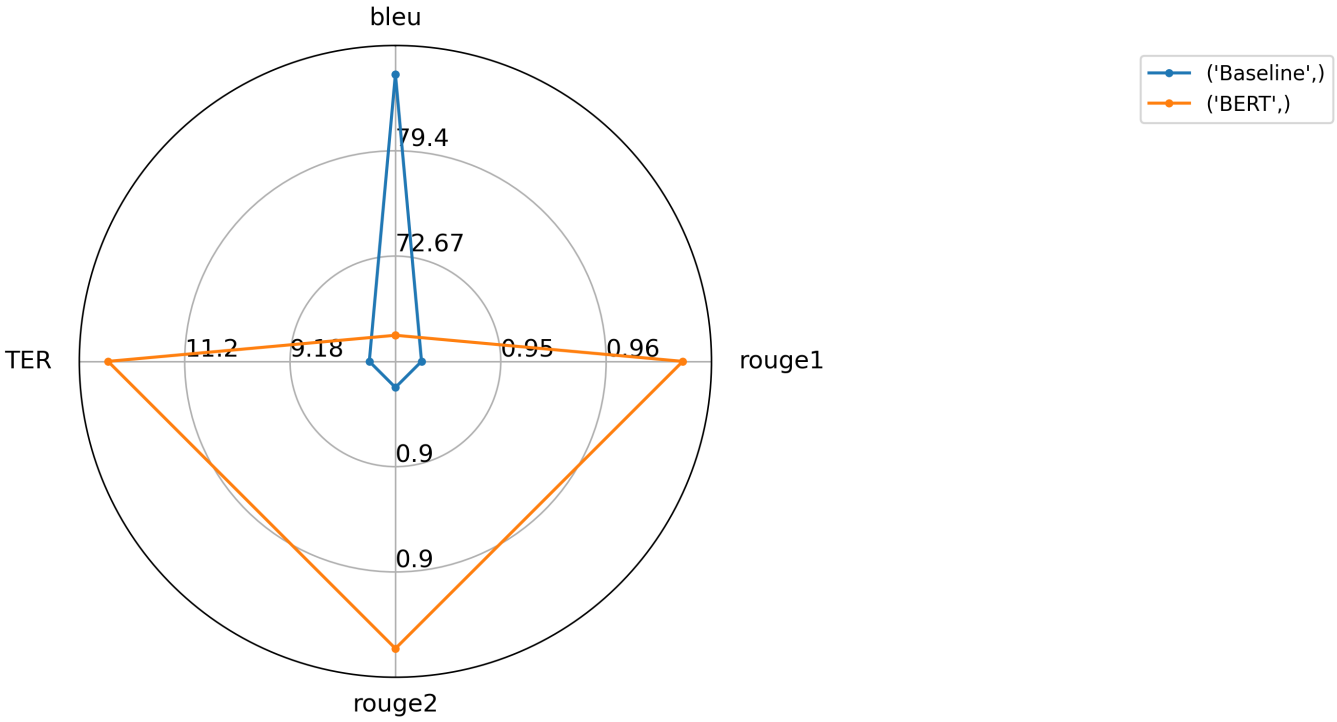
- Batch size for all models: 16
- LR
  - T5: 2e-4
  - BERT 1e-5
- Weight decay for all models: 0.01

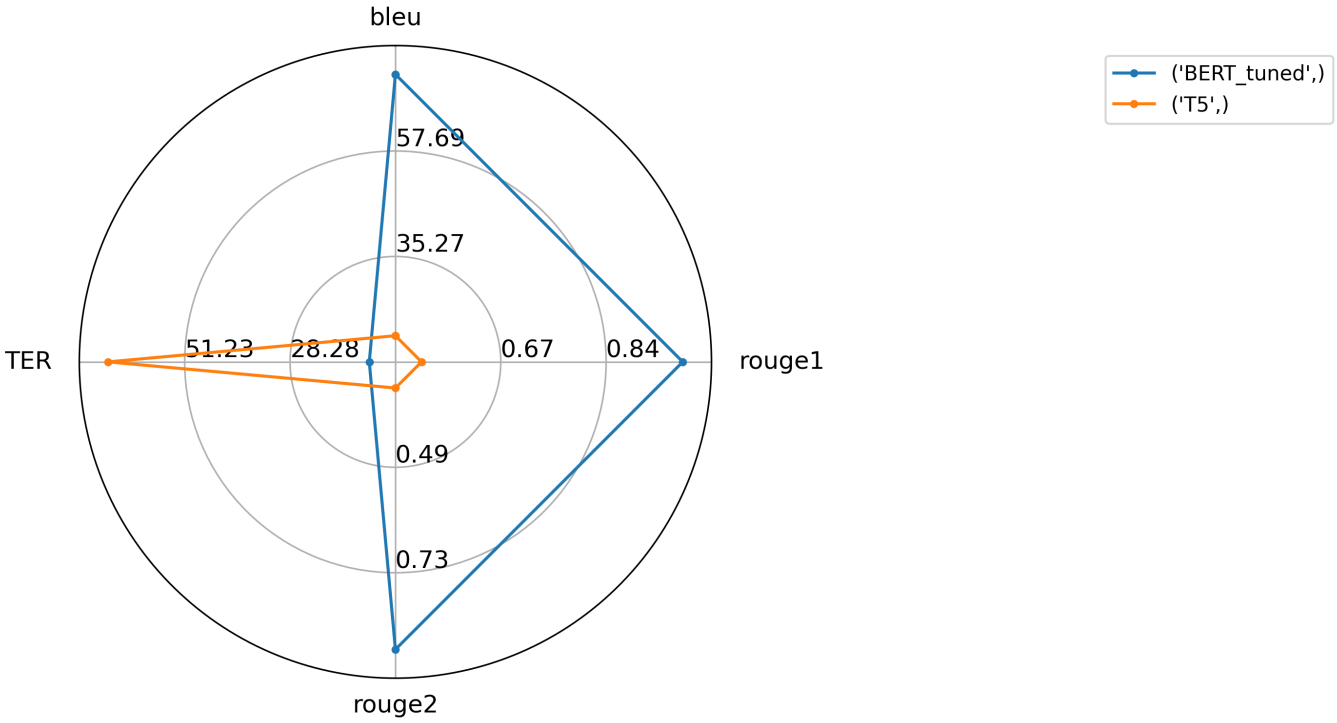
Evaluation

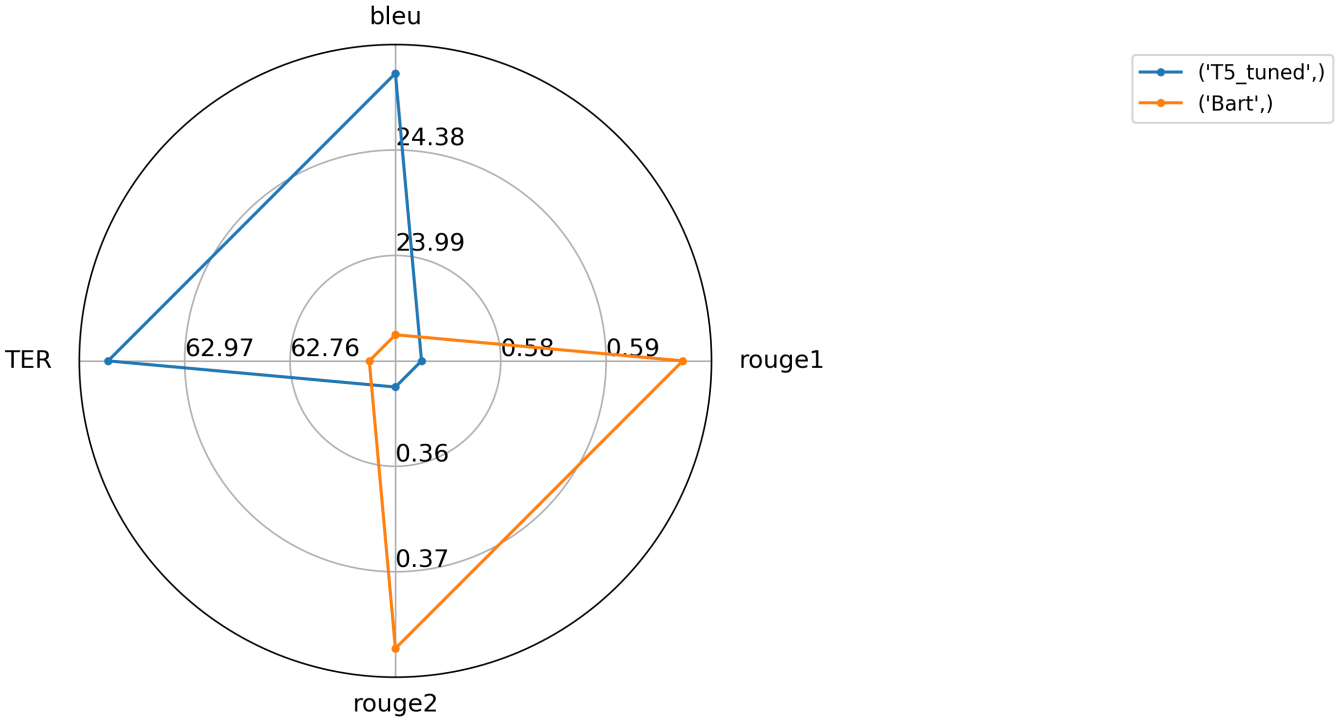
My test dataset consists of 1000 entries. For the evaluation I used the following metrics:
BLEU (Bilingual Evaluation Understudy) is an algorithm for evaluating the quality of text which has been machine-translated from one natural language to another. Quality is considered to be the correspondence between a machine’s output and that of a human: “the closer a machine translation is to a professional human translation, the better it is” – this is the central idea behind BLEU. BLEU was one of the first metrics to claim a high correlation with human judgements of quality, and remains one of the most popular automated and inexpensive metrics.
ROUGE, or Recall-Oriented Understudy for Gisting Evaluation, is a set of metrics and a software package used for evaluating automatic summarization and machine translation software in natural language processing. The metrics compare an automatically produced summary or translation against a reference or a set of references (human-produced) summary or translation.
Translation Error Rate (TER) is a metric for automatic evaluation of machine translation that calculates the number of edits required to change a machine translation output into one of the references.

Results

	bleu	rouge1	rouge2	TER
Baseline	84.30	0.951	0.895	7.660
BERT	67.60	0.958	0.907	12.670
BERT_tuned	74.00	0.956	0.909	11.010
T5	18.40	0.547	0.311	67.930
T5_tuned	24.66	0.579	0.356	63.113
Bart	23.70	0.590	0.369	62.610




bleu
57.69
35.27
TER
51.23
28.28
0.67
0.84
rouge1
0.49
0.73
rouge2
('BERT_tuned',)
('T5',)


bleu
24.38
23.99
TER
62.97
62.76
0.58
0.59
rouge1
0.36
0.37
rouge2
('T5_tuned',)
('Bart',)