Final Report

Nikolay Pavlenko
n.pavlenko@innopolis.university

November 5, 2023

1 Introduction

After reading the paper [1] I have decided not to deviate from the models that they have established for text detoxification as they demonstrated very good results against state-of-the-art methods from that time, and I do not believe myself capable of outpacing their achievements for the purpose of a single course assignment. Therefore, my final solution is the CondBERT model retrained on a larger vocabulary, and its performance is evaluated against simple pre-trained CondBERT and paraGeDi models.

The two primary models, paraGeDi and CondBERT, play significant roles in the task of text detoxification. paraGeDi, an extension of the GeDi model, effectively directs language models like GPT-2 to generate text on specific topics while decreasing toxicity. Conversely, CondBERT is a specialized adaptation of BERT tailored for detoxifying text. Its approach involves detecting toxic words, suggesting replacements using BERT, and employing tactics to preserve the original content. I came up with a hypothesis that aims to improve CondBERT by enlarging its dictionary with tokens from the original dataset to enhance its ability to predict text toxicity. These models present distinct methodologies for addressing text detoxification challenges.

2 Data analysis

Before moving forward with the selection of a model for detoxification, I had to analyse the data. The original dataset is a filtered ParaNMT-detox corpus (500K sentence pairs), which contains 2 categorical columns:

- **reference** - sentence to be detoxified
- **translation** - paraphrased version, supposedly less toxic

It also contains 4 numerical columns:

- **ref_tox** - toxicity level of reference text
- **trn_tox** - toxicity level of translation text
- **similarity** - cosine similarity of reference and translation
- **lenght_diff** - relative difference of length between reference and translation

Given the structure of the dataset, my assumption was that we have to find a model that would translate the reference into a non-toxic sentence and then we would compare the resulting translation with the existing one in the dataset.

Investigation into the data has revealed that it has no zero values and several interesting mathematical properties:

	similarity	length_diff	ref_tox	trn_tox
count	577777.000000	577777.000000	577777.000000	577777.000000
mean	0.758469	0.157652	0.541372	0.434490
std	0.092695	0.108057	0.457571	0.458904
min	0.600001	0.000000	0.000033	0.000033
25%	0.681105	0.066667	0.012171	0.000707
50%	0.754439	0.141791	0.806795	0.085133
75%	0.831244	0.238095	0.990469	0.973739
max	0.950000	0.400000	0.999724	0.999730

Figure 1: Characteristics of the original data

I have noticed that the means of **ref_tox** and **trn_tox** are very similar, so it is possible that the split into toxic references and non-toxic translations does not exist in the original data. To rectify that, I create a new dataset that is used by my detoxification models. This new dataset has four features, rather than 6:

- **toxic** contains the **reference** if **ref_tox** is less than **trn_tox**, and the **translation** otherwise.
- **non-toxic** contains the **translation** if **ref_tox** is less than **trn_tox**, and the **reference** otherwise.
- **old_toxicity** contains the **trn_tox** if **ref_tox** is less than **trn_tox**, and the **ref_tox** otherwise.

- **new_toxicity** contains the **ref_tox** if **ref_tox** is less than **trn_tox**, and the **trn_tox** otherwise.

The dataset above can be safely used in detoxification tasks, as it has clearly separated the toxic sentences from non-toxic. Its mathematical properties prove my point:

	old_toxicity	new_toxicity
count	577777.000000	577777.000000
mean	0.940260	0.035601
std	0.100831	0.079399
min	0.500139	0.000033
25%	0.940145	0.000164
50%	0.983842	0.003456
75%	0.997519	0.027242
max	0.999730	0.499494

Figure 2: Characteristics of the modified data

3 Model Specification

Model that is used for my final solution is CondBERT, which is a specialized adaptation of BERT, designed for the task of rewriting the text in a different style. It utilizes a series of steps to identify and replace toxic words in sentences.

1. **Toxic Word Detection**: The model detects toxic words within input sentences by fetching a toxicity score for each word from the dictionary.
2. **Toxic Word Substitution**: It generates possible substitutes for the identified toxic words using BERT, a language model, and reranks these substitutes based on their similarity and toxicity scores.
3. **Adaptive Threshold**: A dynamic threshold is used to determine toxic words, balancing the proportion of marked toxic words within a sentence.
4. **Content Preservation Heuristics**: The model employs various strategies to maintain the meaning of the replaced words. It prioritizes original tokens, reorders replacements based on the similarity of their embeddings to the original words' embeddings, and penalizes tokens with positive toxicities.

5. **Multiple Token Replacement**: It allows BERT to replace a single masked token with multiple tokens using a beam search approach and scores the sequences based on their token probabilities.

6. **Inference Efficiency**: While considering multiple tokens can enhance accuracy, it may also increase inference time. The model aims to balance between complexity and efficiency for practical use.

4 Training Process

To make CondBERT better fit the dataset I have, I have decided to follow the training procedure proposed in [1], though slightly modified. In their training of CondBERT they use a logistic regression function to learn whether a token is toxic or not, using the pre-set vocabulary of words. Instead of creating the vocabulary from 0, in my training process I increment the already pre-trained vocab with all tokens from my dataset. This increases the knowledge model has about different toxic and non-toxic words, and allows it to find better candidates for replacement during the detoxification.

Another necessary modification I made to ensure that the training works, was to modify the number of iterations of the logistic function from 1000 to 10000. An increase in vocabulary size also increased the complexity of logistic regression, and 1000 iterations where insufficient for it to converge.

5 Evaluation

The evaluation is performed separately for pre-trained CondBERT model, for pre-trained paraGeDi model and for CondBERT model trained on the given dataset. To demonstrate differences between the models performing detoxification I made them translate the same sentences:

```
editor_1.translate("you're becoming disgusting.")
```

```
you're becoming disgusting.
'you ' re becoming sanitary .'
```

```
editor_1.translate("well, we can spare your life. ")
```

```
well, we can spare your life.
'well , we can spare their life .'
```

```
editor_1.translate("monkey, you have to wake up.  ")
```

```
monkey, you have to wake up.
'. , you have to wake up .'
```

Figure 3: Translations of pre-trained CondBERT

```
editor.translate("you're becoming disgusting.")
```

```
you're becoming disgusting.
'you ' re becoming sanitary .'
```

```
editor.translate("well, we can spare your life. ")
```

```
well, we can spare your life.
'well , we can spare their life .'
```

```
editor.translate("monkey, you have to wake up.  ")
```

```
monkey, you have to wake up.
'. , you have to wake up .'
```

Figure 4: Translations of fine-tuned CondBERT

```
[ ]: paraphrase("you're becoming disgusting.")

[ ]: "You'll get disgusted."

     + Code   + Markdown

[ ]: paraphrase("well, we can spare your life.")

[ ]: 'Well, we can save you.'


     paraphrase("monkey, you have to wake up.")

[ ]: 'Monkey. You have to wake up.'
```

Figure 5: Translations of pre-trained paraGeDi

For further For evaluation purposes we will be using metrics from the aforementioned paper: **J**-score, **ACC**, which were implemented in the PMLDL_Assignment1/detox/emnlp2021/metr file. The comparison with the existing dataset will be conducted through the **ACC** metric, as we cannot afford to manually gather the data on the toxicity of reformulated sentences by the model from many different people.

Unfortunately, calculation of the metrics as performed in that file takes an unexpectedly significant amount of time and due to time constraints their results will not be present in the current version of the report.

6 Results

The re-trained CondBERT has proven to be at least as effective at detoxification task as the pre-trained model, and considering that it has a larger vocabulary, it will perform better in general. A comparison between CondBERT and paraGeDi translations showcases that while paraGeDi is significantly better at generating human-like text due to its usage of GPT-2, it is worse at strictly identifying toxic words, as in the examples I have shown above it fails to understand that disgusting and disgusted are both not very nice words, and fails to

recognize that monkey can be an insult.

References

[1] Dale D. et al. Text detoxification using large pre-trained neural models //arXiv preprint arXiv:2109.08914. – 2021.