Final Report

Igor Alentev
BS20-RO-01
i.alentev@innopolis.university

1 Introduction

In this work we are gonna present the solution for task of text "detoxification". The task is to present the solution which will paraphrase (or otherwise modify) the given sentence, making it non-toxic and preserving initial meaning. The approach we are going to present is fine-tuning of encoder-decoder transformer architecture.

For instance, we choose pretrained T5 transformer as base model. Then we fine-tune it on a small selection of english paraphrasing corpora. Corpora contains sentence pairs — toxic sentence with non-toxic ideal paraphasing.

The purpose of this work is to demonstrate that it even small amount of data is enough to help well pretrained general-purpose transformer capture specifics of the task. Narrowing down the context of the model or teaching it the new style (e.g. non-toxic) can find a use in a wide variety of applications. As a result, effective tuning of the model with small amount of resources can be incredible advantage of such kind of models.

The quality of solution will be tested against with baseline algorithmic solution. Moreover, solution will be compared with SOTA model, based on BART architecture.

2 Datamining

Primary sources of the information were:

1. ParaNMT filtered corpus
2. Glove embeddings
3. WordNet
4. Jigsaw bad words dataset

2.1 ParaNMT

While the dataset was considered as a dataset of the toxic sentences and their non-toxic paraphrases with toxicity ranking from people, it seems to be slightly wrong.

It seems that paraphasing is the reverse-translation of the sentence, which was done artificially as well.

Moreover, original work for toxicity ranking proposed one more model.

As a result, the main dataset was artificially generated, artificially ranked and in general is not a very good source of data.

2.1.1 Filtering the filtered

To achieve decent results we can carefully pick the samples from the provided dataset. For instance, experimentally we deduced that filtering the samples which lead from high level of toxicity to the very low in result, filters many insane examples.

Moreover, such filter is beneficial in terms of actually successfully detoxifying the text, while preserving the high similarity of the reference and translation.

Final parameters for filtering:

$$\text{tox}_{\text{ref}} > 0.99 \wedge \text{tox}_{\text{trn}} < 0.01 \qquad (1)$$

2.1.2 Splitting

After filtering the data, we can deterministically sample the data with specified random seed, to achieve reproducible results. For instance, we want 3000 samples for test data. 10000 of samples will be split in $9 : 1$ correspondence for train/validation datasets.

While larger samples are computationally feasible, they tend not to improve final results greatly, as a result, we will stick with these values.

3 Metrics

3.1 Toxicity

To objectively compare the performance of the models in text detoxification we need some toxicity metric. As such, we are going to use ***Detoxify*** package.

Each model predictions will be ranked in the interval $[0, 1]$ for further comparison.

3.1.1 Semantic similarity

Since we have a task of translation (style change) with preservation of the meaning, we might want to use some related metrics. Generally, we have decided to stick with metrics introduced in initial draft.

Final results will be compared using the following metrics:

- BLEU
- METEOR
- Cosine Similarity (GLoVe embedded)

4 Methodology

4.1 Baseline

Baseline was selected realtively low. Algorithm proposed is not from ML family. Simple substitution algorithm finds toxic sequences in the given sentence and replaces then with synonymous sequence.

4.1.1 Datamining

To build the dictionary of pairs (toxic sequence, safe alternative) we took the dataset of toxic words. Each element of the dataset of toxic words was paired with safe synonym generated by WordNet. Generated synonyms were filtered using the same dataset to avoid forbidden words and cyclic toxic replacements.

4.1.2 Replacement

We performed regexp replacement of toxic sequences. Dataset went minimal preprocessing. Moreover, we replaced only complete match of the sequences, without substring match.

4.1.3 Evaluation

While it was proposed to results of baseline model, dropping the unchanged sentences, further research proven that it is bad approach. Indeed, dropping unchanged sentences, we increase the overall toxicity metrics. Moreover, we would need to drop such sentences across all other models, which would result in inconsistencies across the test set. As a result, metrics were further tweaked for a proper representation of the results.

4.2 SOTA Model

As a main model to compare with, we will use current state-of-the-art model in the field. The model was presented in the paper “ParaDetox: Detoxification with Parallel Data”. As a base model was chosen the BART.

4.3 T5-finetuned

As a solution for the task we propose T5 transformer fine-tuned on filtered data. The main advantage of this pretrained T5-small model is it's price(computational)-quality balance. Solution pipeline as follows:

1. Load model checkpoint. In our case it is T5-small
2. Tune the style of the model to the non-toxic. This step can be interpreted in multiple ways. General-purpose transformer can be used as translation model. We can consider it translating from english-toxic to english-non-toxic. That is why we also call it restyling of the model predictions.
3. Evaluate on the test data and note the results

4.4 T5-finetuned^2

After some tweaking of existing results, we came up with an updated solution. Considering we have a **T5-finetuned** model, we can try to further improve it's performance.

For instance, on creation of baseline model, we have derived the dataset of toxic words with their non-toxic synonyms, alternatives. This dataset ideally aligns with our initial dataset of paraphasing. We can concatenate these two datasets to try to further improve the solution. Hence the name — **T5-finetuned^2**.

4.4.1 Fine-tuning

The only difference from the previous approach is that the datasets of baseline model and ParaNMT filtered corpus were concatened before randomly batching into fine-tuning.

5 Evaluation

First and foremost, we might want to consider the difference in performace of our two approaches. **T5-finetuned** and **T5-finetuned^2**. This will help us identify whether the additional training is beneficial for the task.

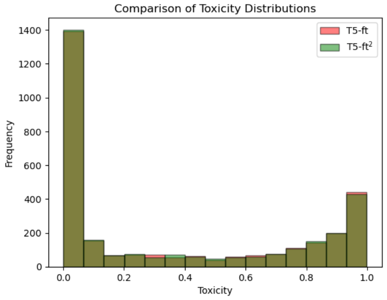

Figure 1: Toxicity distribution

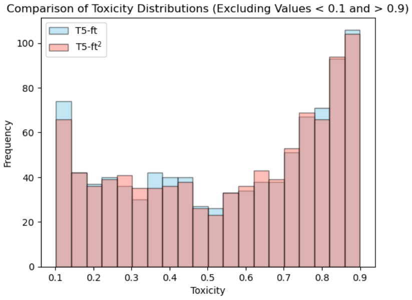

Figure 2: Toxicity distribution (limited)

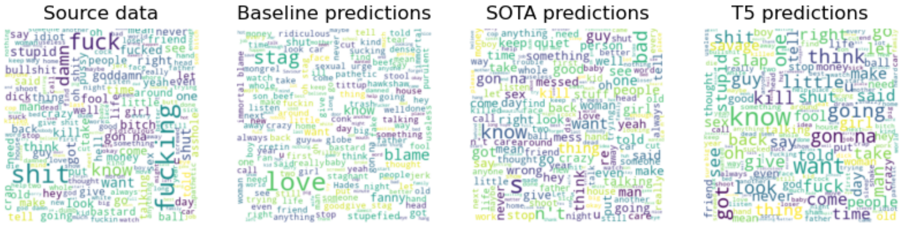

Figure 3: Wordclouds comparison

As we can derive from the toxicity distributions, models are practically identical and no any significant improvement in performance was introduced. Moreover, additional training added more computational and time overhead. As a result, we might want to further consider only the first approach, without additional training.

We can conclude, that the issue with the **T5** model as base is not in it's fine-tuning, rather than the capabilities of the model itself. Extending the tuning dataset will not improve it's capabilities and other approach in future is required.

Further on, we can consider first the wordcloud of the corresponding models. We can see that the source data has a lot of swear words, which are very frequent. Wordcloud is quite good representation for this type of data.

If we consider **baseline** predictions on the very same figure, we can see that predictions are not perfect either. While the word "love" is considered the most frequent, on the second place there is "stag", which is, most probably, used in a toxic context. Moreover, we can find a lot of evidently swear words in the cloud.

Completely the other side of a coin represents the **SOTA** approach. It is difficult to find swear word in the cloud, even if any of them are there. This is a very good result, as it completely removed the open toxicity.

Somewhat in the middle is the **T5-finetuned** approach. While it generally has great results and cloud lacks such dense distribution of bad words, it did not eliminate all of them. At least they are not so densely populated as in the baseline solution.

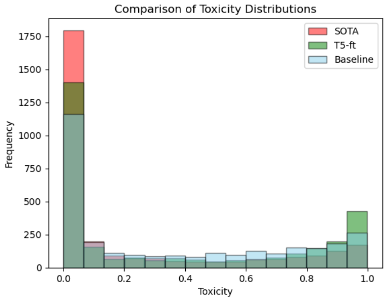

Figure 4: Toxicity distribution

5.1 Toxicity

We can analyze the toxicity distribution of all three models and consider the reasons of such differences. First of all, we can notice that toxicity distribution is approximately the same. It means that the sentences the models struggle most are most probable relatively the same. Some of models just perform a bit better and some of them a bit worse.

Evidently, **SOTA** technique has

the best score in the distribution close to zero, meaning it is the best approach to clean up the sentences. As well as the **T5-finetuned** is the second place. However, we can see that **T5-finetuned** struggles the most on difficult sentences, as the number of sentences with toxicity close to one is the largest. Most probably, baseline solution is better in right most distribution because is better at detecting the bad words in senteces, but not necessarily better at preserving the meaning. The reason for this, baseline simply matches the swear words, while **T5-finetuned** can be easilly fooled by difficult and confusing formulations.

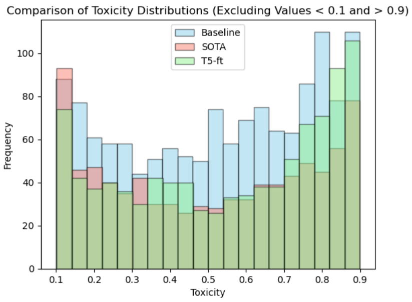

Figure 5: Toxicity distribution (limited)

A bit better representation of the distribution can be seen on the figure 6. Limiting the x axis of the figure for edge cases, we can see comparison of average models performance on the data.

Surprisingly, we can see, that **T5-finetuned** and **SOTA** have pretty similar distributions just like they were same models with slightly different effectiveness. However, **baseline** is definitely just bad in this case. On a larger test set this difference could have been even larger.

5.2 Semantic similarity

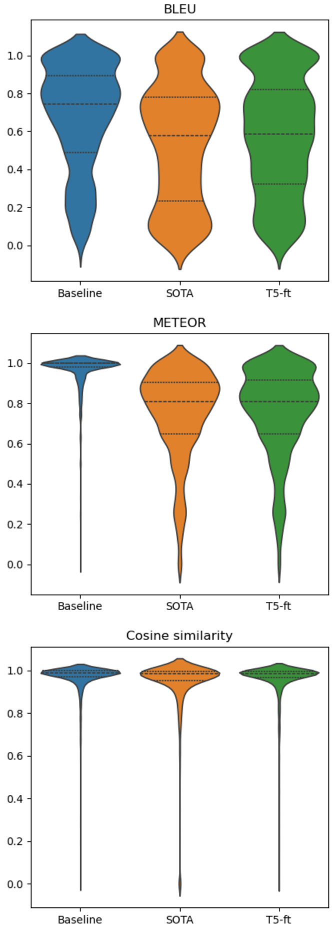

Figure 6: Semantic comparison

Analysis of semantic similarities between the source and prediction can lead to interesting results as well. For this we will use violin plots, representing the distribution of the certain value in the dataset. Finally, we will add lines, denoting quartiles of the data distribution, as it will become useful in the future.

For instance, very expected result is the victory of **baseline** model over all other metrics. The key to success is the cherrypicking of the word it replaces, leaving everything as it was before. As a result, overall score becomes incredibly large, since the sentences practically identical. Even unsuitable synonyms can not affect the overall result of the model. However, as we saw previously, this result comes at cost of the worst detoxification results and meaning violation.

More interesting conclusions we can make comparing the **T5-finetuned** with the **SOTA** algorithm. The overall patterns of the result again confirm the suspicious similarity of the models. Overall shape of the distribution, as well as quartiles generally reflect the same shapes of predictions.

Moreover, **T5-finetuned** tends to have slightly better semantic scores than the **SOTA** model. The reason for this is the preservation of the overall structure of the sentence. While being seemingly built around the similar model, **SOTA** is slightly better at paraphrasing, such as “dickheads” becomes “bad guys”.

Therefore, every metric comes at a cost. **T5-finetuned** is better at preserving the initial structure of the sentence, but worse in detoxification. Conversely **SOTA** approach is better at detoxification, due to non-preservation of initial structure.

6 References

All references provided in original README.md

Original repository with all the data.