Text Detoxification

Almaz Dautov
Innopolis University

Introduction

Text detoxification refers to the process of cleansing or purifying text content to make it more suitable for various purposes, such as enhancing readability, removing offensive or harmful language, or preparing data for natural language processing tasks. This can involve tasks like profanity filtering, content summarization, paraphrasing, or even removing irrelevant information. Text detoxification is crucial for creating a more inclusive and respectful online environment, as well as for improving the quality of text-based data used in machine learning models, sentiment analysis, and other language processing applications. It plays a vital role in ensuring that text content is both safe and effective in achieving its intended goals.

Dataset

Main dataset consists of two datasets ParaNMT filtered and Paradetox.

ParaNMT filtered
The dataset is a subset of the ParaNMT corpus (50M sentence pairs). The filtered ParaNMT-detox corpus (500K sentence pairs)

ParaDetox: Detoxification with Parallel Data (English)
This repository contains information about Paradetox dataset -- the first parallel corpus for the detoxification task -- as well as models and evaluation methodology for the detoxification of English texts. The original paper "ParaDetox: Detoxification with Parallel Data" was presented at ACL 2022 main conference.

Data analysis

For the primary dataset, we retain only the references where the translated toxicity score is lower than that of the references.

Metrics

Metric for evaluation of the model's performance is the J metric, which is the multiplication of sentence-level style accuracy, content preservation, and fluency. Style accuracy (ACC) is measured with a pre-trained toxicity classifier. Content preservation (SIM) is evaluated as the similarity of sentence-level embeddings of the original and transformed texts computed by the model of Wieting et al. (2019). Fluency (FL) measured with the classifier of linguistic acceptability trained on the CoLA dataset (Warstadt et al., 2019). J is computed as the average of their sentence-level product. [1]

3.1 Transfer accuracy (ACC)

Given an output sentence ¯sj and a target style j, a common way of measuring transfer success is to train a classifier to identify the style of a transferred sentence and report its accuracy ACC on generated sentences (i.e., whether ¯sj has a predicted style of j). 14 of 23 surveyed papers implement this style classifier with a 1-layer CNN (Kim, 2014). However, recent large Transformers like BERT (Devlin et al., 2019) significantly outperform CNNs on most NLP tasks, including style classification. Thus, we build our style classifier by fine-tuning RoBERTa-large (Liu et al., 2019) on all our datasets, leading to significantly more reliable ACC evaluation.7

3.2 Semantic similarity (SIM)

A style transfer system can achieve high ACC scores without maintaining the semantics of the input sentence, which also motivates measuring how much a transferred sentence deviates in meaning from the input. 15 / 23 surveyed papers use n-gram metrics like BLEU (Papineni et al., 2002) against reference sentences, often along with self-BLEU with the input, to evaluate semantic similarity. Using BLEU in this way has many problems, including (1) unreliable correlations between n-gram overlap and human evaluations of semantic similarity (Callison-Burch et al., 2006), (2) discouraging output diversity (Wieting et al., 2019), and (3) not upweighting important semantic words over other words (Wieting et al., 2019; Wang et al., 2020). These issues motivate us to measure semantic similarity using the subword embedding-based SIM model of Wieting et al. (2019), which performs well on semantic textual similarity (STS) benchmarks in SemEval workshops (Agirre et al., 2016).

3.3 Fluency (FL)

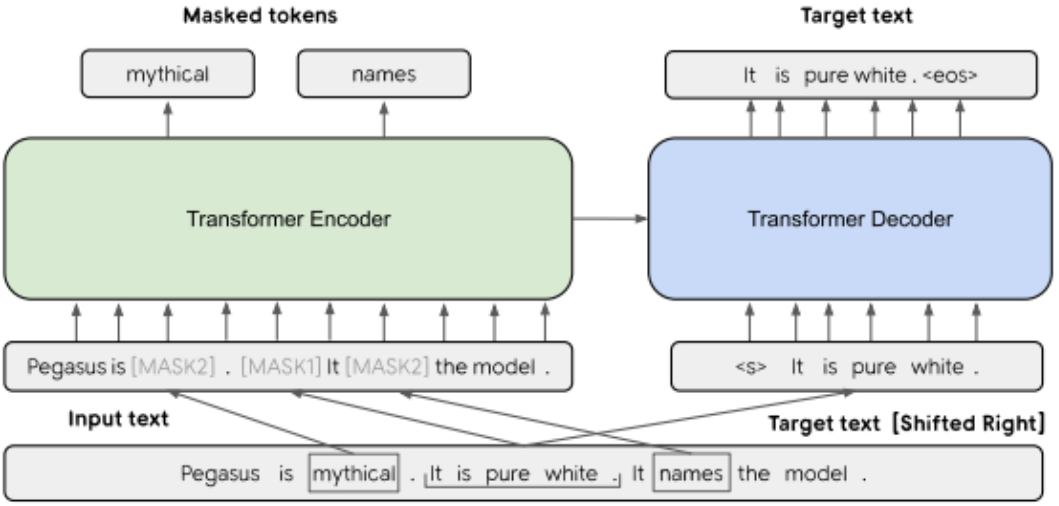
A system that produces ungrammatical outputs can still achieve high scores on both ACC and SIM, motivating a separate measure for fluency. Only 10 out of 23 surveyed papers did a fluency evaluation; 9 of which used language model perplexity, which is a poor measure because (1) it is unbounded and (2) unnatural sentences with common words tend to have low perplexity (Mir et al., 2019; Pang, 2019). To tackle this we replace perplexity with the accuracy of a RoBERTa-large classifier trained on the CoLA corpus (Warstadt et al., 2019), which contains sentences paired with grammatical acceptability judgments. In Table 1, we show that our classifier marks most reference sentences as fluent, confirming its validity. [2]

$$\text{J(ACC, SIM, FL)} = \sum \frac{ACC(X) * SIM(X) * FL(X)}{|X|}$$

Model

The base architecture of PEGASUS is a standard Transformer encoder-decoder.In PEGASUS, the model generates important sentences that have been removed or masked

from an input document as a unified output sequence. This process resembles the extraction of key content, similar to an extractive summary. [3]



Training

The main dataset was divided into three subsets for training, validation, and testing. A small portion, 5%, was allocated for testing, while the remaining 95% was used for training and validation. Within the training set, 20% was set aside for validation. The model was trained with a learning rate of 2e-5, incorporating a weight decay of 0.01, over the course of 5 training epochs.

Epoch	Training Loss	Validation Loss	Acc	Sim	Fl	J	Blue
1	0.392100	0.357948	0.714656	0.750274	0.906254	0.494035	0.554488
2	0.365300	0.344334	0.748804	0.754590	0.908101	0.520794	0.563554
3	0.354200	0.337801	0.761951	0.755998	0.909038	0.531385	0.566634
4	0.345600	0.335123	0.768151	0.757607	0.909578	0.536959	0.569053
5	0.343400	0.334158	0.772030	0.757325	0.909855	0.539474	0.569136

Evaluation

ACC	SIM	FL	J	BLEU
0.7727	0.7600	0.9131	0.5421	0.5710

Results

Original	pegasus
i gotta have her i fell n shit	I have to have her.
I just want a glass of water. Oh, shit!	I just want a glass of water.
We have a big fuckin' problem.	we have a big problem.
Some freak just sent me this.	someone sent me this.

9. References

[1] Text Detoxification using Large Pre-trained Neural Models, David Dale et. al.
[2] Reformulating Unsupervised Style Transfer as Paraphrase Generation, Kalpesh Krishna et. al.
[3] PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization, Jingqing Zhang et. al