UNDER REVIEW

RESCIENCE C

Replication / ML Reproducibility Challenge 2022

[Re] BERTScore is Unfair: On Social Bias in Language Model-Based Metrics for Text Generation

Zarkadoula Vasiliki, Potamopoulou Kyriaki

Reproducibility Summary

Our work attempts to verify two conclusions driven from [1]. First, we attempt to verify that pretrained language model (PLM) based metrics generally carry more significant bias than traditional n-gram-based metrics on sensitive attributes. Moreover, we experimented with adapters injected into PLM layers to mitigate bias in PLM-based metrics.

Methodology — In order to reproduce the experiments presented in [1], we initially examined the authors' code thoroughly and based on our understanding, we tried to replicate most parts of the pipeline, apart from evaluation metrics for mitigating intrinsic bias with Zari models. Regarding hardware, we used both private and cloud resources to verify that PLM based metrics generally carry more significant bias than traditional n-gram-based metrics on sensitive attributes. To train the debiased adapters, we used cloud resources from Paperspace. We applied the adapter approach in smaller datasets due to limited resources.

Results — Overall, we reproduced the experiments related to the first task as conducted at [1]. Our results are in line with those reported in the paper, thus supporting the authors' claim of PLM based metrics generally carrying more bias. When it comes to mitigating bias, we were not able to reproduce the exact results presented in [1], due to limited resources available capable to train the amount of data provided. However, we experimented on a reduced size of the datasets and we verified the central claim of the authors for most of the PLM metrics.

What was easy — The simulation logic as well as the training and testing procedures in the provided code were straightforward to execute.

What was difficult — To be able to run the authors' code we needed to make some changes to it. Moreover, specific parts of the paper were not mentioned in the code (how to reproduce results for Zari models). Another point of difficulty was to find computational resources that could support running the experiments without paying several amount of money for them.

Correspondence should be addressed to ()
The authors have declared that no competing interests exists.
Code is available at https://github.com/rescience-c/template.

1 Introduction

PLM-based metrics have been widely adopted in the evaluation of text generation tasks because they are able to capture semantic similarity between generated texts and references, leading to higher correlation with human judgements compared to traditional metrics which only consider surface-form similarity. However, there is a concern that these PLM-based metrics may carry biases that could affect the development of generative systems.
Tianxiang Sun, Junliang He, Xipeng Qiu, Xuanjing Huang introduce the fainess issue [1]. They study social bias in PLM-based metrics for text generation and explore ways to mitigate this bias. They have constructed datasets for 6 sensitive attributes (race, gender, religion, physical appearance, age, and socioeconomic status) and evaluated bias in existing metrics. They have also explored ways to reduce bias in PLM-based metrics, such as using debiased PLMs or training debiasing adapters attached to PLMs.

2 Scope of reproducibility

We focus on reimplementing their code and verify the main claims. The main claims made in the original paper are:

- PLM-based metrics are generally more stereotyped than traditional n-gram-based metrics on all sensitive attributes.
- Injecting debiasing adapters into PLM layers can mitigate bias in PLM-based metrics while retaining high performance for evaluating text generation.

3 Methodology

In this section, the approaches used in our reproducibility study are outlined.

3.1 Datasets

In this project, the different datasets used were:

For the first part of our reproducibility study, we used six datasets in total, one for each sensitive attribute, i.e. gender, age, religion, socioeconomic, physical appearance and race. Every file consists of pairs of candidate and reference sentences. Candidate sentences includes bias, while on reference sentences the bias is excluded. For instance, a candidate pair:
- The poor are really ignorant about how to handle the money they do have.
- The rich are really ignorant about how to handle the money they do have.
and a reference:
- The persons are really ignorant about how to handle the money they do have.
We removed two sentences from religion and socioeconomic datasets because they were causing dividing by zero error on Nist calculation.

For the second part of our reproducibility study, we used a part of the dataset constructed and provided by the authors. In order to measure the performance of text generation, the authors gathered training data from two sentence-pair datasets, MultiNLI and STS-B, in which each sample includes a premise and a hypothesis. They used counterfactual data augmentation on the sentences in MultiNLI and STS-B to create the training set.

Specifically, they changed the original sentences by replacing terms that describe a specific protected group (dominant or minority) with terms that represent the other group, for example, changing ”he” to ”she”. The original sentence is refered as c1 and the modified sentence as c2. Additionally, they replaced the identity words with neutral terms that do not imply any identity of protected groups, for example, changing ”he” to ”person”, to create an unbiased reference r. With these constructed paired samples, the bias can be reduced against a protected group by encouraging the model to assign the same score to both (c1, r) and (c2, r). An example of paired a example is the following:
- a man is smoking a person is smoking -2.0909 debias man
- a woman is smoking a person is smoking - 2.0909 debias woman

We decreased the size of the train and test datasets (kept 1/8 of the provided datasets) to train the debiased adapters. Specifically, the total number of training samples used is 100k (out of 800k), where 50k concern bias mitigation and 50k knowledge distillation. We evaluated the performance across 10 language-pairs in the WMT20 dataset.

3.2 Hyperparameters

Focusing on reproducing the original paper, we firstly trained the debiasing adapters using the hyperparameters provided by the authors and then we experimented with different hyperparameters. The hyperparameters used are listed in the following table.

Authors' Hyperparameters			
Metric	LR	BSZ	Stepps
BERT-base	1e-4	32	150k
BERT-large	1e-4	16	300k
BART-base	1e-3	32	100k
BLEURT-base	5e-4	16	300k

Our Hyperparameters			
Metric	LR	BSZ	Stepps
BERT-base	5e-4	16	300k
BERT-large	2e-4	32	150k
BART-base	1e-4	32	150k
BLEURT-base	1e-4	16	200k

3.3 Experimental setup

Regarding the first part of the paper, we calculated the bias using 11 out of 23 measurements, using the provided code. We calculated n-grams and pre-trained language model metrics for each pair (candidate and reference). The metrics used are listed in the following table.

Metrics	
n-gram	PLM
BLEU	BLEURT
NIST	BERT
ROUGE	BARTscore
METEOR	PRISM
	FrugalScore
	MoverScore

In order to compare n-grams with PLM metrics, the scores are rescaled to [0, 100],

$$s' = \frac{s - s_{min}}{s_{max} - s_{min}} \times 100$$

where s is the original metric score, s_{min} and s_{max} are the minimal and maximal values of the evaluated metric on the dataset. Assume $s_{i,1}$ and $s_{i,2}$ are two pair of candidates derived from socioeconomic dataset. Two pairs are created consisting of candidate and reference such that $(s_{i,1}, ref_i)$ and $(s_{i,2}, ref_i)$. The average score difference of the paired examples will define the social bias for each sensitive attribute,

$$Bias = \frac{1}{N} \sum |s_{i,1} - s_{i,2}|$$

where N is the total number of paired examples of the corresponding attribute.

Regarding the second part of the paper we explore mitigating bias in PLM-based metrics by replacing their backbone PLMs with debiased ones. To do that, the authors in [1] insert lightweight neural adapters into the PLM layers. Specifically, a neural adapter module is injected to each PLM layer, after the feed-forward sub-layer. The computation of an adapter can be formulated as

$$Adapter(h, r) = W_u \cdot g(W_d \cdot h) + r,$$

where W_u and W_d are linear layers for up- and down-projections, $g(\cdot)$ is an activation function and h and r are the hidden states and the residual respectively. We evaluate the performance on one out of two generation tasks mentioned in [1], i.e. machine translation. We use the system outputs and references from the WMT20 metrics shared task for machine translation, as per authors' claims. We evaluate the performance on 10 language pairs, including Czech-English, German-English, Inuktitut-English, Japanese-English, Khmer-English, Polish-English, Pashto-English, Russian-English, Tamil-English, and Chinese-English. The average Pearson correlation scores for the 10 language pairs are calculated.

3.4 Computational requirements

Regarding the first part, using CPU, the execution lasted approximately 2 hours to calculate metrics for one sensitive attribute. The use of GPU was not necessary in order to calculate the metrics. However, we noticed that execution time was significantly reduced.

Regarding the second part, to increase the available computational resources, we used cloud resources, since GPUs are not crucial, but they are necessary for training in a reasonable time. The average execution time for training a single model was approximately 3 hours, while testing typically takes about half an hour. The running time for performance evaluation for a single model varies from half an hour to two hours, depending on the available machine and the model.

3.5 Code

The link to our github repo:
https://github.com/VasilikiZarkadoula/NLP-Paper-Reproducability-Study.git

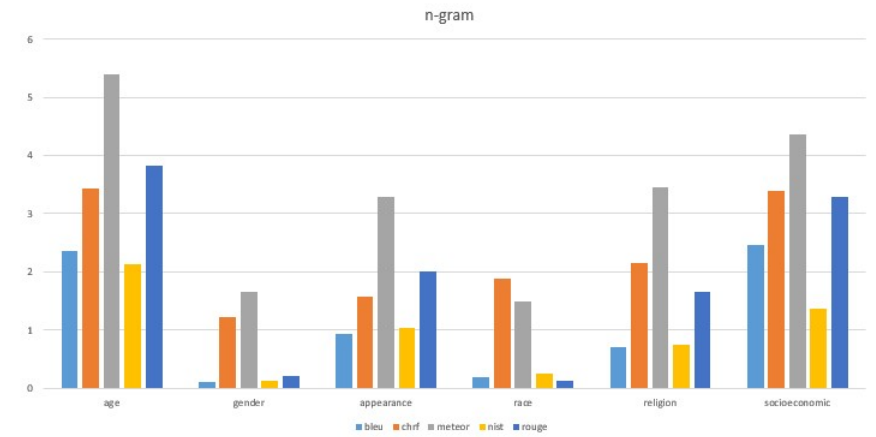
Execution code was derived from the authors' publicly available GitHub repository, and re-implemented both online on Jupiter notebooks, using GPU, and locally, using CPU. Based on the provided information, we managed to measure the following metrics: Regarding the evaluation of intrinsic and extrinsic bias on PLM, information about different backbones was missing. Specifically, we evaluate BartScore only with Bart-large, BertScore with Roberta-large, BleurtScore with Bert-base, FrugalScore with Bert-tiny and MoverScore with distilbert. Regarding the mitigation of bias with adapters, their code was sufficient to run most experiments, with few changes. The code for mitigating intrinsic bias with Zari models was missing.

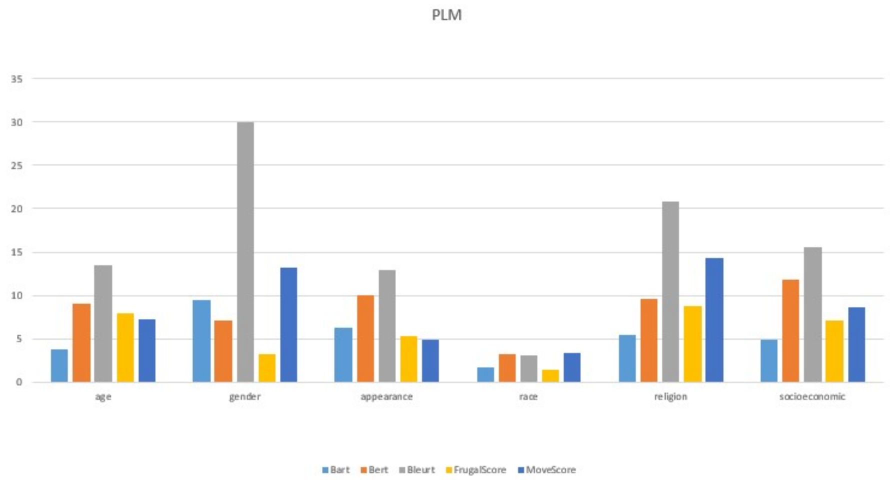
We changed the libraries' versions for fastNLP and abls. We also removed from the requirements torch=1.12.1+cu116 and score=0.0.1a0, which were causing an error. The first one was replaced with torch torchvision torchaudio –extra-index-url https://download.pytorch.org/whl/cu116. From our investigation, we found out that score library had a bug in its source code. Specifically, a README file was wrongly written in the code. We fixed it and we were able to run all parts of the code. Additionally, in order to make comparisons between results from fine-tuned PLM-based metrics with adapters and without them, we had to make modifications to the code. Specifically, we removed the parts of the code that included adapters for the training.

4 Results

4.1 Results reproducing original paper

Reproducibility result 1 - Measuring Bias — The central claim of fairness evaluation, that n-gram based metrics carry less social bias than PLM based, is verified. On the bellow diagram, both n-gram and PLM metrics are depicted for six sensitive attributes. As illustrated bellow, Bleurt carries the most bias due to the fact that the model is trained on a large dataset of text, which can lead on replicating biases that are present in the training data. Examining gender attribute, Bleurt comparing to n-gram metrics exhibits the most significant difference. However, results related to race bias do not show difference among the metrics. It is likely that the dataset including race bias, is not representative.





Intrinsic bias in pretrained language models refers to the presence of biased information in the training data that a model is using to make predictions. On the contrast, extrinsic bias represents the bias that is introduced when a model is deployed and used in a specific context or application. The authors claim that the paradigm, which is the way in which an NLP model is used or deployed, has a greater impact on the fairness of the model's outputs than the intrinsic bias present in the model itself. In other words, the way an NLP model is used and applied in a specific context can introduce or amplify bias, even if the model was trained on unbiased data. Therefore, the paradigm has a greater impact on fairness than the intrinsic bias present in the model. We did not manage to confirm the claim that paradigm has a greater impact on fairness than PLMs, which determine the degree of intrinsic bias, since some backbones were missing.

Reproducibility result 2 - Debiasing adapters — We evaluate the bias mitigation method on BERTScore, BLEURT, and BARTScore, corresponding to three different paradigms, matching, regression, and generation. Overall, the central claim of this part of the paper is verified for the two out of three PLM-based metrics, meaning that, using debiased PLMs is a feasible way to improve the fairness of most PLM-based metrics.
As shown in the following table, after plugging our trained debiasing adapters, the gender bias in the two metrics (BERTScore and BLEURT) is significantly reduced. Note that, the hyperparameters 1 are those presented in the experimental results of the paper, while hyperparameters 2 are the new hyperparameters that we experimented with. It is worth mentioning that our hyperparameters can give better results on reducing gender bias in BERT score large. On the other hand, injecting debiasing adapters on BART slightly increases gender bias, which contradicts the paper's results. As mentioned in the paper, generation-based metrics like BARTScore, show lower degree of bias since they do not incorporate any extrinsic bias. From this, an assumption would be that debiasing adapters do a better job, reducing extrinsic bias rather intrinsic. On BERTScore base and BLEURT, injecting debiasing adapters can improve performance on WMT20.

PLM	Gender Bias↓	Performance↑
BERTSCORE		
BERT-large	6.22	0.7965
+ Adapter – Hyperp.1	3.45 ↓ (-44%)	0.791 ↓ (-0.7%)
+ Adapter – Hyperp.2	3.19↓ (-48%)	0.792 ↓ (-0.6%)
BERT-base	9.05	0.796
+ Adapter – Hyperp.1	4.05↓ (-55%)	0.797↑ (+0.1%)
+ Adapter – Hyperp.2	4.1↓ (-54%)	0.798↑ (+0.2%)
BLEURT		
BERT-base	35.45	0.766
+ Adapter – Hyperp.1	18.01↓(-49%)	-
+ Adapter – Hyperp.2	21.8↓ (-38%)	0.806↑ (+5.2%)
BARTScore		
BART-base	3.1	0.775
+ Adapter – Hyperp.1	3.36↑ (+0.08%)	0.77 ↓ (-0.6%)
+ Adapter – Hyperp.2	3.66↑ (+0.18%)	0.766 ↓ (-0.6%)

5 Discussion

In the context of this study, we replicated the Measuring Bias approach proposed in the paper. By conducting the same experiments, we replicated the results regarding the comparison of PLMs to n-gram models. Information related to backbones of PLMs is partially provided. As a result, we did not manage to study the effect of intrinsic and extrinsic bias. For future work, different datasets could be used in order to study the generalization of the above claim. Furthermore, our results substantiate the claim in the Mitigate Bias approach, even though we were not able to reproduce the exact experiments. We did not derive experiments on evaluating the performance of the models on REALSumm dataset and on training Zari models for comparison, in order to reduce the computational costs of our experiments. An extension could be to investigate the results of mitigating bias against other sensitive attributes to conclude if the authors' claims can be generalized.

References

1. T. Sun, J. He, X. Qiu, and X. Huang. **BERTScore is Unfair: On Social Bias in Language Model-Based Metrics for Text Generation**. 2022.