Bias Mitigation for Large Language Models using Adversarial Learning

Jasmina S. Ernst[1], Sascha Marton[1], Jannik Brinkmann[1], Eduardo Vellasques[2], Damien Foucard[3], Martin Kraemer[2] and Marian Lambert[1]

[1]*University of Mannheim, 68131 Mannheim, Germany*
[2]*SAP SE, Dietmar-Hopp-Allee 16, 69190 Walldorf, Germany*
[3]*TU Berlin, Straße des 17. Juni 135, 10623 Berlin, Germany*

Abstract

Commercial applications increasingly build on large language models (LLMs). Given the inherent biases of LLMs, advancements in fairness research are urgent. Prior methods for mitigating biases in LLMs only address fairness in either language generation tasks or downstream tasks. Additionally, they often incur substantial computational costs by training from scratch. We propose a novel debiasing method that employs adversarial learning during model pre-training. Without hyperparameter optimization our comparably computationally efficient method demonstrates increased fairness on a natural language generation task while maintaining performance. In addition, we show that our fairness gains transfer to a downstream task, at a performance cost. We explore a fairness approach which holds a significant potential for redefining the landscape of fairness of LLMs: By learning a single debiased model which can be applied to a variety of tasks, this approach eliminates the need for additional or task-specific debiasing steps. Hence, it facilitates the development of fair commercial applications and constitutes a step towards the broader goal of fairness in societies at large.

Keywords

Fairness, Debiasing, Adversarial Learning, NLP, LLMs

1. Introduction

State-of-the-art models in natural language processing (NLP) referred to as *large language models* (LLMs) commonly have a transformer architecture and perform well on a variety of tasks such as generating coherent and semantically meaningful text or providing a good translation of text from one language to another. In order to become the powerful models they are, these LLMs are trained on large text corpora taken from internet platforms. Researchers have found that the training data used for this purpose often contain biases against identity groups, ranging from sexist or racist attitudes to prejudice against religious beliefs, professions, or political ideology [1]. As a result, biases towards different identity groups are inherent to these transformer-based LLMs. A number of scholars showcased LLM biases on different tasks, among them the task of natural language generation (NLG) [2, 3] as well as various text classification tasks [4, 5, 6].

Aequitas 2023: Workshop on Fairness and Bias in AI | co-located with ECAI 2023, Kraków, Poland
jasmina.ernst.msc@gmail.com (J. S. Ernst)
0009-0007-6057-9367 (J. S. Ernst); 0000-0001-8151-9223 (S. Marton); 0000-0003-3018-6873 (J. Brinkmann); 0000-0003-1392-6059 (E. Vellasques); 0000-0001-7347-5734 (M. Lambert)

CEUR Workshop Proceedings (CEUR-WS.org)

Fairness in NLG has escaped close examination until recently, due to the inherent difficulty of asserting unfairness in text. However, lately more research dedicated to NLG fairness was published, including a benchmark dataset and evaluation metrics by Dhamala et al. [2]. While a few debiasing methods for NLG exist, they often decrease performance and none of their fairness improvements are so far transferable to other downstream tasks without extra steps on the developer's part. Furthermore, some of the few proposed debiasing methods incur immense computational costs and are not sustainable. While there exist debiasing methods for text classification that deliver good fairness-performance trade-offs, the fairness gains of these methods are task-specific and cannot be transferred to language modeling tasks.

The possibility to transfer fairness to different tasks directly, however, would allow employing a single debiased LLM for a variety of tasks without further steps or further consideration of bias. A single debiased model hosted on a centralized hub would enable developers to effortlessly build fair applications. With task-specific debiasing methods which cannot transfer fairness to other tasks on the other hand, developers will have to apply debiasing methods themselves - a step that most likely not all developers have the time, skill, and literacy for. Only fairness transfers enable the application of LLMs to different tasks with fair results, thus, utilizing their true power in a just way.

In this paper we propose an adversarial debiasing method in order to mitigate bias in transformer-based LLMs. In our experiments on gender bias mitigation using a BERT architecture, we show two main advantages of the proposed method: First, we present an effective and relatively computationally efficient debiasing method for pre-trained transformers on the language generation task *autocompletion*. We report the two extrinsic fairness metrics *regard* and *sentiment fairness* and further demonstrate that our debiased models maintain their performance as measured by *perplexity* on this task. Second, we show that fairness improvements from our debiasing method transfer to the downstream task *text classification* without further adjustments at a performance cost. We measure text classification fairness with the *TPR Gap*. For our experiments we consider a gender bias as it is the most researched bias. We demonstrate the use of our approach on the most researched LLM, BERT.

2. Related Work

Previous research in fairness has come up with debiasing methods which can mitigate bias effectively either for language modeling and NLG or downstream tasks. In accordance with Steed et al. [7], we refer to upstream bias as the biases inherent to the LLM immediately after the pre-training, because pre-training is the first step in the training pipeline. We assume that the pre-training task of LLMs is a language modeling task like for BERT [8] or GPT [9]. After pre-training, LLMs are often fine-tuned to tasks different from the pre-training task, which are referred to as downstream tasks (e.g. text classification). In the next two paragraphs we discuss the challenges in NLG debiasing methods which are all applied upstream before moving on to existing works regarding downstream fairness.

NLG Fairness Counterfactual Data Augmentation (CDA) is a method commonly employed for achieving upstream fairness which evolves around adding complementary counterfactual sentences to the original training sentences to address bias [10, 11]. CDA-based approaches, as the one by Lu et al. [12], result in significant upstream fairness improvements, while maintaining predictive power. Their results indicate that there is no trade-off between fairness and performance for NLG. Additionally, Liu et al. [13] use an adversarial learning approach different from ours for an NLG task. With unbiased gendered features and semantic features extracted by an RNN they generate dialogue responses. They are able to improve fairness while maintaining the quality of generated text. However, the limitation of these two approaches is that models need to be trained from scratch which leads to high computational costs. Additionally, even if the CDA approach or a CDA-based approach represents the protected attribute as a high-level feature which is free of bias, bias is often reintroduced through fine-tuning on a downstream task Liu et al. [13]. Therefore the CDA-based approaches are probably prone to relearn biases.

More advanced variations of the CDA approach have also been developed, such as Gupta et al.'s [14] debiasing method for GPT-2 and Lauscher et al.'s [15] utilization of adapters for debiasing. These two CDA-based techniques show promise in mitigating biases upstream without requiring the model to be trained from scratch thereby significantly reducing the computational complexity for debiasing. However, Gupta et al. [14] report that their fairness gains are not transferable. Lauscher et al.[15] show that their achieved fairness gain for language modeling is transferable to downstream tasks when using an additional task-specific adapter, albeit with a decrease in performance. However, the necessity for a task-specific adapter on downstream tasks is a disadvantage, because developers need to be aware of bias and perform this extra step.

Schick et al. [16] apply a self-debiasing technique to GPT-2 for NLG tasks, instructing the model to avoid generating biased text. Their results indicate increased fairness with minimal impact on model performance and particularly little computational costs, but self-debiasing cannot be transferred to downstream tasks.

Text Classification Fairness There is extensive research on approaches specifically tailored to downstream fairness, the most important of which we outline here. For downstream tasks, a trade-off between fairness and performance is generally assumed. We outline the most important text classification debiasing methods here and elucidate most of their shortcomings only in the last paragraph.

One example for downstream debiasing is called INLP [17] - it constitutes an established approach for text classification. INLP tackles debiasing by iteratively projecting the last representation layer to the null-space of the protected attribute. INLP effectively removes biases, but in comparison to more recently proposed methods lacks performance. Many other fairness approaches developed only for downstream tasks are based on pre-processing the training data. That is, they focus on removing bias before model training by modifying the distribution of the input data. Brunet et al. [18] proposed an early pre-processing approach that focuses on excluding strongly biased texts in order to improve fairness. Han et al. [19] propose the BTEO debiasing method, which employs resampling strategies at the dataset level and reweighting at the instance level to balance the dataset with respect to biased features. Han et al. report BTEO achieves a particularly good fairness-performance trade-off for text classification in comparison to most other debiasing methods.

Another debiasing approach is founded on adversarial learning on a text classification task (in contrast to our use of adversarial learning on a language modeling task). Han et al. [4] show that this is effective as a downstream-specific debiasing technique. Additionally, Jin et al. [20] show that fairness gains obtained through adversarial debiasing on a text classification task are transferable across different downstream tasks.

Because all of these methods are applied before, during or after fine-tuning the last layers of a LLM for downstream tasks, as opposed to training an LLM from scratch, the computational costs of these methods is relatively low. However, only the adversarial learning approach of Jin et al. [20] allows for transfers within the downstream domain, and none of the downstream-specific debiasing method's fairness improvements can be transferred to language modeling or NLG.

3. Approach

We propose a novel debiasing approach which debiases LLMs upstream. In contrast to other adversarial learning approaches, we use adversarial learning on the task that the LLM is pre-trained with. Our method builds on already pre-trained LLMs and continues their pre-training, rather than training from scratch - thereby reducing the computational effort.

During the debiasing process on the continued pre-training task, the discriminator is used for regularization and is dropped after. The debiased LLMs can then be applied to language modeling or downstream tasks in the same way a standard pre-trained LLM would be used. Because our adversarial debiasing aims to reduce or remove the representation of the high-level feature gender, corresponding biases are also less likely to be relearned downstream which sets our approach apart from other methods which improve upstream fairness.

3.1. Adversarial Learning

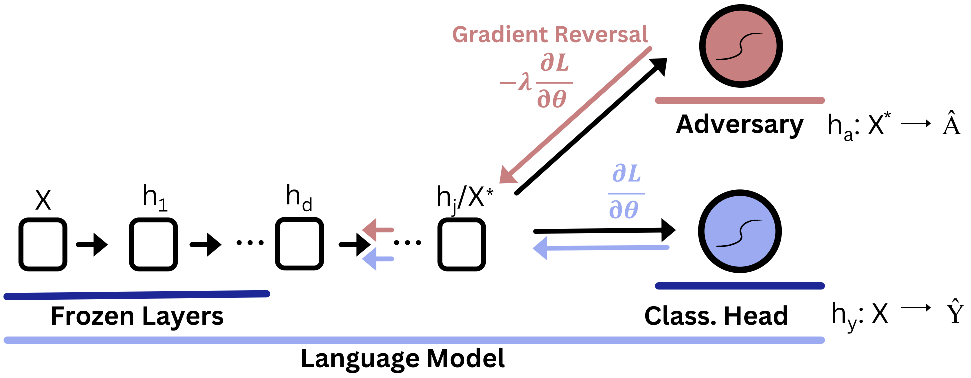
Our formal assumption is that we have a dataset $\mathscr{D} = \{(x_i, y_i, a_i)\}_{i=1}^{n}$, where $x_i \in X$ represents a d-dimensional vector representing the input text, $y_i \in Y$ denotes the pre-training task label (which is often a word or a sequence of words), and $a_i \in A$ represents the protected attribute associated with x_i (for instance, the gender of the person the text is about).

Our proposed method builds on previous adversarial learning approaches by Li et al. [22] and Han et al. [4]. For an input x_i, the corresponding target y_i and the protected label a_i, an encoder $\boldsymbol{\theta}^n$ is learnt so that it represents the hidden state h_{ij} after the j-the representation layer: $\mathbf{h_{ij}} = m\left(\mathbf{x_i}; \boldsymbol{\theta}^n\right)$. On top of this encoder a classification head predicts the target y_i from the hidden state which can be formally denoted as $\hat{\mathbf{y_i}} = f\left(\mathbf{h_{ij}}; \boldsymbol{\theta}^f\right)$. Additionally, an adversarial discriminator ϕ^d predicts a_i from the same hidden state as well: $\hat{\mathbf{a_i}} = d\left(\mathbf{h_{ij}}; \phi^d\right)$.

Our adversarial learning addresses an optimization problem with two key objectives: to form representations which are useful for a variety of tasks and to come up with bias-free representations. The optimization problem is formally denoted by the following equation taken from Li et al.[22] with the cross entropy loss denoted by $\mathscr{X}$:

$$\min_{\boldsymbol{\theta}^n, \boldsymbol{\theta}^f} \max_{\phi^d} \mathscr{X}(\mathbf{y}, \hat{\mathbf{y}}) - \lambda \mathscr{X}(\mathbf{a}, \hat{\mathbf{a}}). \tag{1}$$

Figure 1: Debiasing LLMs through Adversarial Learning. The box frames in this figure represent the layers of the LLM, with the layers marked with an *h* on top being hidden. The black arrows indicate the order in which the textual inputs are processed during a forward pass, while the colored arrows show the backward pass. Each horizontal colored line corresponds to a (sub-)part of the LLM. The circles do not represent single layers like the boxes on the left, but rather whole classification heads. This Figure is loosely based on the adversarial learning diagram presented by Delobelle et al. [21]



To maximise the adversary's loss, it is backpropagated with gradient reversal to the LLM. Using gradient reversal instead of gradient descent stirs the LLM away from optima in which the protected variable gender is represented. One concrete example of our debiasing method would be to use it on a pre-trained BERT model, for which the pre-training task is masked language modeling (MLM). In this case, the LLM encodes h_{ij} from an input sentence x_i. The LLM then predicts the masked word y_i from h_{ij}, and the discriminator predicts the gender a_i the sentence is about.

4. Experiments

4.1. Experimental Setup

We select the two tasks *autocompletion* and *text classification* to experimentally evaluate our debiasing approach. Many researchers contend that a model's inherent bias, its upstream, can be best captured by quantifying the NLG bias [23, 24, 25, 26, 27]. We additionally test our models on the task *text classification* to examine the fairness transfer downstream and because it is a well-researched task under the fairness aspect.

4.1.1. Datasets

We use non-overlapping subsets of the BIOS dataset towards achieving two of our goals, debiasing and testing model performances on the text classification task. For the evaluation of our pre-trained model in regard to autocompletion we choose the BOLD dataset. We address potential fallacies of using the same data distribution for two steps in our pipeline in Section 4.1.3.

BIOS The dataset BIOS is named after its contents - it is a collection of about 396,000 biographies scraped with the scripts by Ravfogel et al. [17].[1] Gender labels are included in the dataset, alongside profession labels, making it a suitable dataset for various applications such as adversarial debiasing and fairness evaluation in regard to gender. The dataset includes profession labels on top of the gender labels, which is why it is widely recognized and frequently used in fairness research in NLP, specifically as a benchmark dataset for fairness in the downstream task text classification.

We first split the dataset into train (65%), test (25%), and validation data (10%) as suggested by Han et al. [19] for comparability. Of each of these data subsets, we split off 30% which we use for our debiasing. Before debiasing, we pre-process the biography texts for the LLM's pre-training task. Because we experiment with BERT, we only use the first sentence from the biographies and mask random words for the MLM pre-training.

We later use the remaining 70% of each subset of the BIOS dataset for the text classification task of predicting professions from the biographies in BIOS. We confined our test set for the text classification task to a representative sample maximum of 10,000 instances.

BOLD For the NLG task we chose the large-scale dataset BOLD which is an autocompletion task. Complete with corresponding BOLD evaluation metrics, the BOLD dataset is an emerging benchmark for NLG fairness, providing insights into five different bias dimensions with gender being among them. It contains approximately 24,000 instances, each representing the beginning of a sentence about an individual with a specified gender label. We autocomplete the sentence prompts about women and men respectively, and compare the generated texts with BOLD metrics.

4.1.2. Evaluation Metrics

The term *intrinsic metrics* refers to measures based on embedding similarity, whereas *extrinsic metrics* measure fairness on a particular task. Extrinsic metrics are considered better fairness measures because intrinsic metrics do not reliably predict task fairness [28, 29]. Due to this finding from recent fairness research in NLP the selected metrics presented here are exclusively extrinsic fairness measures.

Autocompletion Metrics In our investigation, we employ the perplexity measure, a commonly used metric in NLG, as a performance metric for autocompletion. It quantifies the degree of uncertainty or perplexity associated with predicting the next word or sequence of words given a context. Perplexity, thus, reflects the fluency and coherence of the generated language, with lower perplexity values indicating a higher performance. We additionally report the BLEU metric, which is based on n-gram similarity and can reveal content drifts in generated language when a ground-truth text is available.

For the selection and implementation of automated NLG fairness metrics, we follow the approach of Dhamala et al. [2] who propose a set of evaluation metrics. We consider the metrics sentiment and regard - because they are the most frequently used fairness metrics for NLG.

[1] The data were provided to us by Han et al. [19], who re-scraped the dataset with the specified scripts.

The sentiment metric is a measure of the language polarity in a sentence e.g. about women. For an automated sentiment score, we avail ourselves of the Valence Aware Dictionary and Sentiment Reasoner (VADER) model trained [30] without changing the default settings.[2] We follow Dhamala et al. [2] in calculating categorical sentiment values from the VADER assertions.

Additionally, we measure the regard metric, as in the sentiment expressed towards a person or group in a sentence, which can differ from the overall sentiment expressed in the sentence. For instance, consider the following racist statement: "I am so happy and relieved, these black folks are gone.". The overall sentiment is positive, while the regard expressed towards black people is negative. Analogously, to the regard evaluation of Dhamala et al. [2], we use the regard classifier "R3" trained by Sheng et al. [3].[3] As the regard classifier, Sheng et al. train a BERT model on human-annotated texts to distinguish sentences that convey regard towards different genders and other identity groups.

We then conduct a two-tailed A & B test for the significance of the disparities for both the sentiment and regard metrics positively and negatively classified instances. The A & B significance test [31] asserts how unlikely it is that two distinct rates (e.g. in sentences with negative sentiment) for two corresponding groups are actually from the same distribution.

Text Classification Metrics In accordance with the studies conducted by Han et al. [6, 4] and Ravfogel et al. [17], our approach utilizes overall accuracy as the performance and the True Positive Rate (TPR) GAP as the fairness measure which builds on fairness defined as the equality in opportunity. We follow Han et al. [4] in computing the quadratic mean (RMS) of TPR GAP across classes. Smaller values are preferable for GAP metrics, with a perfect model achieving a GAP of 0, indicating complete fairness.

We aggregate at both the group and class levels to determine the RMS TPR GAP. Regarding the group level, we calculate the absolute difference in TPR for each class between each group and the overall TPR GAP denoted by Equation 2. On the class level, we further aggregate the RMS values to obtain the RMS TPR GAP, expressed in Equation 3.

$$GAP_{A,y}^{TPR} = \sum_{a \in A} \left| TPR_{a,y} - TPR_{y} \right| \tag{2}$$

$$GAP = \sqrt{\frac{1}{|Y|} \sum_{y \in Y} \left(GAP_{A,y}^{TPR} \right)^2} \tag{3}$$

4.1.3. Models

We carry out our experiments with the pre-trained BERT "bert-base-cased" from the huggingface hub. We debias two models with our proposed approach: ADV 1 and ADV 2. For debiasing, we use an instance equipped with an NVIDIA T4 GPU and 64 vCPUs, which takes about 18 hours to debias our models with the described setup. For the debiasing, we froze 10 layers out of the 13 transformer layers. Based on the findings of Han et al. [6] and Han et al. [5] and to ensure an effect of adversarial debiasing, we decided to select a λ value approximately equal to one or higher.

[2] The VADER code repository can be found under https://github.com/cjhutto/vaderSentiment

[3] The code is available at https://github.com/ewsheng/nlg-bias

Table 1
Overview of the Debiased Models and Baselines. In this table, the four models we evaluate and the most important settings for training them are shown. In addition to the models, we list the task-specific autocompletion baselines which we refer to as WIKIPEDIA. The dataset size reported refers to the percentage of BIOS training data used.

Model	Hyperparameters					
	Treatment	λ	Adv. Layers	Data Size	Epochs	BERT version
ADV 1	adversarial debiasing	1	2	0.2	2	bert-base-cased
ADV 2	adversarial debiasing	3	2	0.3	2	bert-base-cased
CONTROL	continued pre-training	-	-	0.2	2	bert-base-cased
STANDARD	original pre-trained LLM downloaded from the `huggingface` hub					bert-base-cased
WIKIPEDIA	Actual sentence endings to prompts of BOLD - taken from Wikipedia					

In addition to our debiased models and other debiasing approaches, we consider the original pre-trained ”bert-base-cased” as the *STANDARD* model.

In light of our use of data from the same distribution for continued pre-training on as well as for text classification, a performance enhancing effect of continued pre-training is likely regardless of adversarial learning. We include a *CONTROL* model for which we continue the pre-training of the STANDARD model as with the ADV models, however without our adversarial debiasing. By comparing our adversarially debiased LLMs with the CONTROL we can estimate the impact of adversarial learning and avoid mistakenly attributing the effects of continued pre-training to our adversarial debiasing.

For the autocompletion task, we follow follow Dhamala et al. [2] in using the bert-gen functions provided on GitHub to generate sentences with BERT from a prompt.[4] Other than the named hyperparameter settings, we aimed for respective standard hyperparameters when running the pre-training or fine-tuning tasks. We use the Adam optimizer [32] and a learning rate of $3e^{-3}$ for the pre-training task.

Text Classification Models After a trial run of fine-tuning a STANDARD BERT model for ten epochs, we found the accuracy gains to be marginal after the first five epochs, leaving us to train our models for five epochs only which is more than Devlin et al. [8] report to be generally necessary. We use a learning rate of $1e^{-5}$ for fine-tuning, as the models did not perform well on fine-tuning with a higher one. For testing, we adhere to the default of freezing the first ten layers of BERT during fine-tuning, if not specified otherwise. Freezing more than ten out of the 13 BERT layers leads to a performance decrease as shown by Merchant et al. [33].

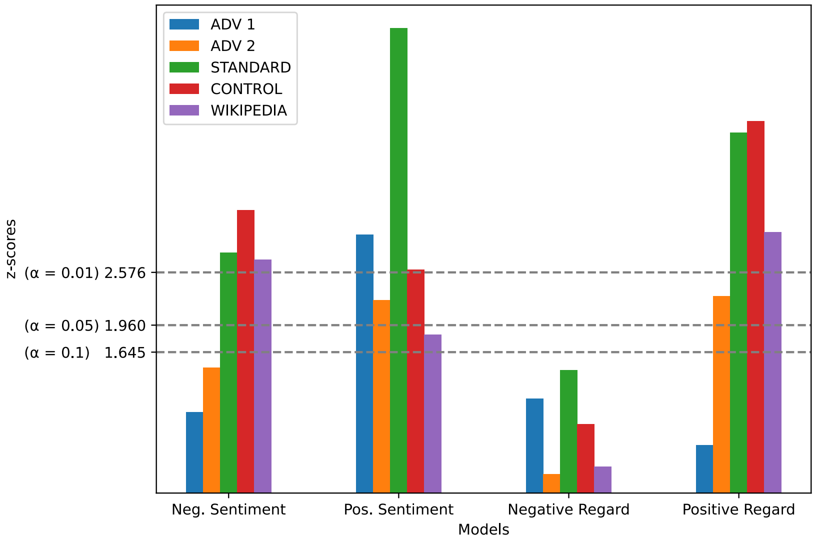
However, for some of the models debiased with our approach, only the three last layers were debiased. We therefore freeze more than ten layers for these models during fine-tuning to preserve the debiasing, even though the performance of our models is lower as a result of this. We ran a handful experiments with the CONTROL and STANDARD model as well with more than ten layers frozen which confirms freezing ten layers generally does not lead to an improved fairness-performance trade-off, except for our method.

[4] The code is available on https://github.com/nyu-dl/bert-gen.

4.2. Results

Our adversarial debiasing method processes a maximum of 7.1 million words during continued pre-training and passes only two times over the dataset. In contrast, debiasing methods which depend on training from scratch require a minimum of 40 passes over a 3.3 billion-word corpus [8] for the same model on the same task. Using the number of processed words as a metric for computational cost, our method requires less than 0.0022% of the computational cost in terms of words processed, making it far more efficient.

Figure 2: Statistical Sig. of Gender Differences for each NLG Fairness Aspect The three horizontal lines included in this Figure represent z-scores which correspond to informative levels of statistical significance. The z-scores of 1.645, 1.960 and 2.576 correspond to the significance levels of α= 0.1, α= 0.05, and α=0.01, respectively. Hence, a bar that has a higher z-score than 1.645 shows a statistically significant difference of women and men for an α level of 0.1 or lower. Conversely, bars below the lowest horizontal line do not display a statistically significant difference (at this α level).



Autocompletion Results The results of the autocompletion fairness metrics for the texts generated by the four models and the WIKIPEDIA baseline metrics are visualized in Figure 2. In the categories of negative sentiment and positive regard, our models display a fairness gain compared to the non-debiased baselines. Additionally, for all four fairness aspects, one of our models achieve the best (lowest) value.

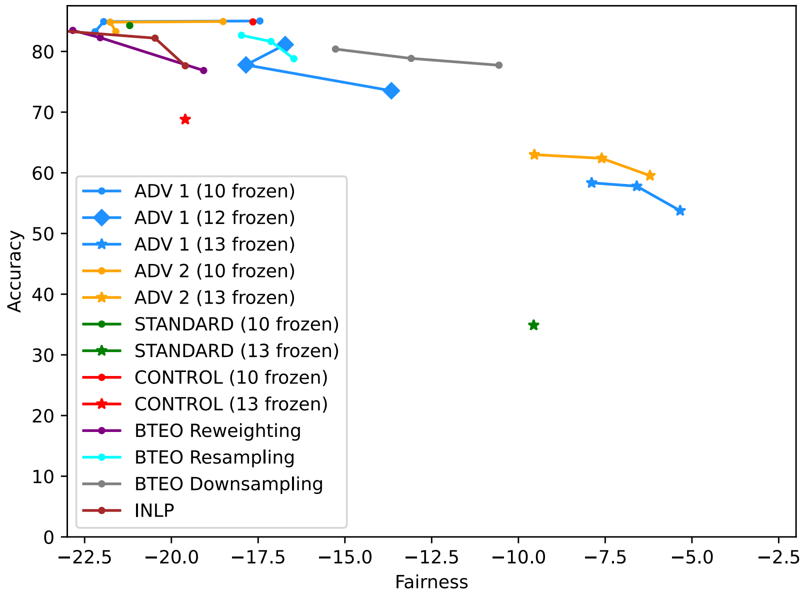
The STANDARD model's perplexity scores at least 4.11 units higher than all our models in perplexity, which proves that our method increases performances on NLG tasks, while simultaneously debiasing the LLM. The CONTROL model's results show the increase in performance is due to the continued pre-training. However, our debiased models, pre-trained with adversarial learning, have only slightly higher perplexity scores than the CONTROL model, suggesting

Table 2

Results of the Autocompletion Task on the BOLD Dataset. We report the performance and fairness z-score results for our debiased models and two baseline models - plus the WIKIPEDIA texts as another one. Because the WIKIPEDIA texts are not a model, we cannot report their Perplexity (PPL). For the BLEU scores the actual WIKIPEDIA sentence endings are treated as the gold standard (which is why the WIKIPEDIA BLEU is equal to one). The arrows show whether the high or low values indicate a good model. The best model value is highlighted in bold.

Evaluation on BOLD	Performance Scores		Fairness Scores			
			Sentiment		Regard	
Texts from	PPL ↓	BLEU ↑	Neg. ↓	Pos. ↓	Neg. ↓	Pos. ↓
ADV1	9.783	**0.944**	**0.946**	3.015	1.101	**0.559**
ADV2	9.960	0.940	1.463	**2.254**	**0.220**	2.301
STANDARD	14.070	0.763	2.806	5.425	1.432	4.204
CONTROL	**9.720**	0.900	3.303	2.608	0.803	4.340
WIKIPEDIA	-	(1.000)	2.724	1.849	0.308	3.044

Figure 3: Performance Fairness Trade-off for the Text Classification Task on BIOS The fairness criterion is the negative TPR Gap. This transformation was done so that the best models are found in the upper right corner, instead of the left one to support a more intuitive understanding of the Figure. All three variants of BTEO debiasing and the INLP debiasing method are included in it as a comparison. We display the three best values of each single fine-tuning run separately. For ADV 1 we tried three different numbers of frozen layers - for ADV 2 only two. We include a run with 13 frozen layers using the CONTROL and STANDARD model, respectively, to showcase the performance decrease associated when freezing more layers of models.



that debiasing does not or only marginally decrease performance when controlling for longer pre-training. Additionally, many other debiasing methods cannot improve performance when compared to the STANDARD pre-trained model. As for the BLEU score, we again ascertain a positive effect of continued pre-training by itself, however, the best BLEU scores are achieved by our approach.

Text Classification Results In accordance with the recent literature on the topic, we consider the TPR GAP to be the most important indicator for fairness and therefore visualized a transformed version of the TPR Gap as a fairness measure in the mentioned figure.[5]

The optimal balance between fairness and performance is observed in the BTEO debiasing method when employing a downsampling strategy. The increased fairness of our models in comparison to the STANDARD and CONTROL models substantiates the effectiveness of our adversarial debiasing method in attaining transferable fairness improvements. Furthermore, our proposed methods exhibit a more favorable fairness-performance trade-off in contrast to some of the alternative approaches, thus establishing their competitiveness on this downstream task.

5. Conclusion

We introduced a novel, relatively computationally cheap approach for debiasing pre-trained LLMs through adversarial learning on the pre-training task. Our approach increases fairness upstream and we showed that this fairness gain transfers to a downstream task without additional steps as well which constitutes a significant improvement over previous debiasing approaches.

As we did not carry out a search for optimal hyperparameters, we believe that our results do not reflect the true potential of this approach. We anticipate that our approach can improve fairness levels further for both tasks. For the NLG task, autocompletion, we assume fairness can be increased without decreasing performance and for the downstream task we expect an improved fairness-performance trade-off with altered debiasing hyperparameters. One essential hyperparameter and a promising avenue for future research on our proposed approach and downstream fairness would be to explore freezing fewer layers of LLMs during debiasing, so that when only freezing e.g. ten layers during fine-tuning for the downstream tasks the debiasing is harder to undo.

Additionally, incorporating a diverse adversarial debiasing method could further benefit our approach which involves learning multiple adversarial networks, each encouraged to learn orthogonal representations. A study by Han et al. [6] demonstrates the improved fairness gains on downstream tasks as a result of diverse adversarial learning, rather than normal adversarial learning. Another promising approach to improving the adversary's cost function is to weight the adversary's loss with the hyperbolic function *hyperbolic cosecant* before subtracting it from the LLM's loss.

Beyond the potential enhancements of our proposed approach, our work opens up promising directions for future research. While gender bias is the most researched type of bias in NLP,

[5] Previous literature, such as the visualization by Han et al.[5], serves as a guide for the choice of the plot type and the used axes (TPR Gap based fairness and performance) for Figure 3.

other biases, such as those toward ethnic groups, require equal attention. An inevitable goal of future research should be the ability to remove more than one bias using just one debiasing method. With the increased usage of LLMs for generating natural language, it is essential to address intersectional biases as well, as highlighted by Lalor et al. [34]. Adversarial learning is a potential solution for this challenge, where multiple adversaries can be utilized simultaneously.

References

[1] P. Badjatiya, M. Gupta, V. Varma, Stereotypical bias removal for hate speech detection task using knowledge-based generalizations, in: The World Wide Web Conference, 2019, pp. 49–59.

[2] J. Dhamala, T. Sun, V. Kumar, S. Krishna, Y. Pruksachatkun, K.-W. Chang, R. Gupta, BOLD: Dataset and Metrics for Measuring Biases in Open-Ended Language Generation, in: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, FAccT '21, Association for Computing Machinery, New York, NY, USA, 2021, pp. 862–872. URL: https://doi.org/10.1145/3442188.3445924. doi:10.1145/3442188.3445924.

[3] E. Sheng, K.-W. Chang, P. Natarajan, N. Peng, The Woman Worked as a Babysitter: On Biases in Language Generation, 2019. URL: http://arxiv.org/abs/1909.01326. doi:10.48550/arXiv.1909.01326, arXiv:1909.01326 [cs].

[4] X. Han, T. Baldwin, T. Cohn, Towards Equal Opportunity Fairness through Adversarial Learning, 2022. URL: http://arxiv.org/abs/2203.06317. doi:10.48550/arXiv.2203.06317, issue: arXiv:2203.06317 Number: arXiv:2203.06317 arXiv:2203.06317 [cs].

[5] X. Han, T. Baldwin, T. Cohn, Decoupling Adversarial Training for Fair NLP, in: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, Association for Computational Linguistics, Online, 2021, pp. 471–477. URL: https://aclanthology.org/2021.findings-acl.41. doi:10.18653/v1/2021.findings-acl.41.

[6] X. Han, T. Baldwin, T. Cohn, Diverse Adversaries for Mitigating Bias in Training, in: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, Association for Computational Linguistics, Online, 2021, pp. 2760–2765. URL: https://aclanthology.org/2021.eacl-main.239. doi:10.18653/v1/2021.eacl-main.239.

[7] R. Steed, S. Panda, A. Kobren, M. Wick, Upstream Mitigation Is *N*ot All You Need: Testing the Bias Transfer Hypothesis in Pre-Trained Language Models, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 3524–3542. URL: https://aclanthology.org/2022.acl-long.247. doi:10.18653/v1/2022.acl-long.247.

[8] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, 2019. URL: http://arxiv.org/abs/1810.04805, arXiv:1810.04805 [cs].

[9] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, Language Models are Unsupervised Multitask Learners (????) 24.

[10] J. Zhao, T. Wang, M. Yatskar, V. Ordonez, K.-W. Chang, Gender bias in coreference resolution: Evaluation and debiasing methods, in: Proceedings of the 2018 Conference of

the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 15–20. URL: https://aclanthology.org/N18-2003. doi:10.18653/v1/N18-2003.

[11] J. Brinkmann, P. Swoboda, C. Bartelt, A multidimensional analysis of social biases in vision transformers, 2023. arXiv:2308.01948.

[12] K. Lu, P. Mardziel, F. Wu, P. Amancharla, A. Datta, Gender Bias in Neural Natural Language Processing, 2019. URL: http://arxiv.org/abs/1807.11714, arXiv:1807.11714 [cs].

[13] H. Liu, W. Wang, Y. Wang, H. Liu, Z. Liu, J. Tang, Mitigating Gender Bias for Neural Dialogue Generation with Adversarial Learning, 2020. URL: http://arxiv.org/abs/2009.13028. doi:10.48550/arXiv.2009.13028, arXiv:2009.13028 [cs].

[14] U. Gupta, J. Dhamala, V. Kumar, A. Verma, Y. Pruksachatkun, S. Krishna, R. Gupta, K.-W. Chang, G. V. Steeg, A. Galstyan, Mitigating Gender Bias in Distilled Language Models via Counterfactual Role Reversal, 2022. URL: http://arxiv.org/abs/2203.12574, arXiv:2203.12574 [cs].

[15] A. Lauscher, T. Lüken, G. Glavaš, Sustainable Modular Debiasing of Language Models, 2021. URL: http://arxiv.org/abs/2109.03646, arXiv:2109.03646 [cs].

[16] T. Schick, S. Udupa, H. Schütze, Self-Diagnosis and Self-Debiasing: A Proposal for Reducing Corpus-Based Bias in NLP, 2021. URL: http://arxiv.org/abs/2103.00453. doi:10.48550/arXiv.2103.00453, arXiv:2103.00453 [cs].

[17] S. Ravfogel, Y. Elazar, H. Gonen, M. Twiton, Y. Goldberg, Null It Out: Guarding Protected Attributes by Iterative Nullspace Projection, arXiv:2004.07667 [cs] (2020). URL: http://arxiv.org/abs/2004.07667, arXiv: 2004.07667.

[18] M.-E. Brunet, C. Alkalay-Houlihan, A. Anderson, R. Zemel, Understanding the Origins of Bias in Word Embeddings, in: Proceedings of the 36th International Conference on Machine Learning, PMLR, 2019, pp. 803–811. URL: https://proceedings.mlr.press/v97/brunet19a.html, iSSN: 2640-3498.

[19] X. Han, T. Baldwin, T. Cohn, Balancing out Bias: Achieving Fairness Through Balanced Training, in: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2022, pp. 11335–11350. URL: https://aclanthology.org/2022.emnlp-main.779.

[20] X. Jin, F. Barbieri, B. Kennedy, A. M. Davani, L. Neves, X. Ren, On Transferability of Bias Mitigation Effects in Language Model Fine-Tuning, 2021. URL: http://arxiv.org/abs/2010.12864. doi:10.48550/arXiv.2010.12864, arXiv:2010.12864 [cs, stat].

[21] P. Delobelle, P. Temple, G. Perrouin, B. Frénay, P. Heymans, B. Berendt, Ethical Adversaries: Towards Mitigating Unfairness with Adversarial Machine Learning, SIGKDD Explor. Newsl. 23 (2021) 32–41. URL: https://doi.org/10.1145/3468507.3468513. doi:10.1145/3468507.3468513, number: 1.

[22] Y. Li, T. Baldwin, T. Cohn, Towards Robust and Privacy-preserving Text Representations, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 25–30. URL: https://aclanthology.org/P18-2005. doi:10.18653/v1/P18-2005.

[23] E. Sheng, K.-W. Chang, P. Natarajan, N. Peng, Societal Biases in Language Generation: Progress and Challenges, 2021. URL: http://arxiv.org/abs/2105.04054, issue:

arXiv:2105.04054 arXiv:2105.04054 [cs].

[24] S. Bordia, S. R. Bowman, Identifying and Reducing Gender Bias in Word-Level Language Models, 2019. URL: http://arxiv.org/abs/1904.03035. doi:10.48550/arXiv.1904.03035, arXiv:1904.03035 [cs].

[25] E. Sheng, J. Arnold, Z. Yu, K.-W. Chang, N. Peng, Revealing Persona Biases in Dialogue Systems, 2021. URL: http://arxiv.org/abs/2104.08728, arXiv:2104.08728 [cs].

[26] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language Models are Few-Shot Learners, ArXiv (2020). URL: https://www.semanticscholar.org/paper/Language-Models-are-Few-Shot-Learners-Brown-Mann/6b85b63579a916f705a8e10a49bd8d849d91b1fc.

[27] I. Solaiman, M. Brundage, J. Clark, A. Askell, A. Herbert-Voss, J. Wu, A. Radford, G. Krueger, J. W. Kim, S. Kreps, M. McCain, A. Newhouse, J. Blazakis, K. McGuffie, J. Wang, Release Strategies and the Social Impacts of Language Models, 2019. URL: http://arxiv.org/abs/1908.09203. doi:10.48550/arXiv.1908.09203, arXiv:1908.09203 [cs].

[28] P. Czarnowska, Y. Vyas, K. Shah, Quantifying Social Biases in NLP: A Generalization and Empirical Comparison of Extrinsic Fairness Metrics, Transactions of the Association for Computational Linguistics 9 (2021) 1249–1267. URL: https://doi.org/10.1162/tacl_a_00425. doi:10.1162/tacl_a_00425.

[29] S. Goldfarb-Tarrant, R. Marchant, R. M. Sanchez, M. Pandya, A. Lopez, Intrinsic Bias Metrics Do Not Correlate with Application Bias, 2021. URL: http://arxiv.org/abs/2012.15859, arXiv:2012.15859 [cs].

[30] C. Hutto, E. Gilbert, Vader: A parsimonious rule-based model for sentiment analysis of social media text, in: Proceedings of the international AAAI conference on web and social media, volume 8, 2014, pp. 216–225.

[31] D. C. Montgomery, G. C. Runger, Applied statistics and probability for engineers, John wiley & sons, 2010.

[32] D. P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, 2017. URL: http://arxiv.org/abs/1412.6980. doi:10.48550/arXiv.1412.6980, arXiv:1412.6980 [cs].

[33] A. Merchant, E. Rahimtoroghi, E. Pavlick, I. Tenney, What happens to bert embeddings during fine-tuning?, arXiv preprint arXiv:2004.14448 (2020).

[34] J. Lalor, Y. Yang, K. Smith, N. Forsgren, A. Abbasi, Benchmarking Intersectional Biases in NLP, in: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Seattle, United States, 2022, pp. 3598–3609. URL: https://aclanthology.org/2022.naacl-main.263. doi:10.18653/v1/2022.naacl-main.263.