Parameter-efficient Modularised Bias Mitigation via AdapterFusion

Deepak Kumar¹, Oleg Lesota¹, George Zerveas³, Daniel Cohen¹,⁴,
Carsten Eickhoff⁵,⁶, Markus Schedl¹,², Navid Rekabsaz¹,²
¹Johannes Kepler University Linz, Austria
²Linz Institute of Technology, AI Lab
³AI Lab, Brown University, USA george_zerveas@brown.edu
⁴Dataminr, USA daniel.cohen@dataminr.com
⁵University of Tübingen, Germany c.eickhoff@acm.org

Abstract

Large pre-trained language models contain societal biases and carry along these biases to downstream tasks. Current in-processing bias mitigation approaches (like adversarial training) impose debiasing by updating a model’s parameters, effectively transferring the model to a new, irreversible debiased state. In this work, we propose a novel approach to develop stand-alone debiasing functionalities separate from the model, which can be integrated into the model on-demand, while keeping the core model untouched. Drawing from the concept of AdapterFusion in multi-task learning, we introduce DAM (Debiasing with Adapter Modules) – a debiasing approach to first encapsulate arbitrary bias mitigation functionalities into separate adapters, and then add them to the model on-demand in order to deliver fairness qualities. We conduct a large set of experiments on three classification tasks with gender, race, and age as protected attributes. Our results show that DAM improves or maintains the effectiveness of bias mitigation, avoids catastrophic forgetting in a multi-attribute scenario, and maintains on-par task performance, while granting parameter-efficiency and easy switching between the original and debiased models.

1 Introduction

Large pre-trained language models (PLM) and their variations fine-tuned on downstream tasks encode societal biases and stereotypes. A large body of research shows the existence of these biases (Zhao et al., 2019; Sheng et al., 2019; Rekabsaz et al., 2021b), and discuss their potential harms (Blodgett et al., 2020) in various tasks such as occupation prediction from biographies (De-Arteaga et al., 2019), information retrieval (IR) (Rekabsaz et al., 2021a; Rekabsaz and Schedl, 2020), and machine translation (Stanovsky et al., 2019). A common approach to mitigating these biases is to adjust model parameters according to specific fairness criteria, achieved using optimization methods such as adversarial training (Elazar and Goldberg, 2018; Barrett et al., 2019; Han et al., 2021b; Rekabsaz et al., 2021a).

The resulting debiased models significantly diminish the effect of protected attributes (e.g. gender, race, etc.) on model’s predictions. However, depending on the usage context, characteristics of the input or fairness-utility trade-off considerations, system designers or end-users might in practice prefer to instead use the original (potentially biased) model for processing some requests. As an example, while a fairness-aware document retrieval model should tease out gender attributes when processing bias-sensitive queries (Krieg et al., 2022) (like how to become CEO?), other common queries (like earliest pregnancy symptoms) may specifically require gender information to obtain satisfactory retrieval results. Previous studies also show that one may need to occasionally accept slight performance degradation in exchange for realizing higher fairness (Zerveas et al., 2022; Biega et al.,...
We argue that on-demand control over the strength of bias mitigation is crucial for the broad adoption of bias mitigation models. Using existing approaches, this would require maintaining and deploying multiple large parallel models for every protected attribute, in practice resulting in overly complex and resource-heavy pipelines and increased latency.

To reduce this untenable burden, we introduce **Debiasing with Adapter Modules (DAM)**. In our approach, bias mitigation functionalities can be trained in fully decoupled modules from the building blocks solving the main task (see Figure 1). These debiasing modules are learned independently, and can be selectively inserted into the model to deliver debiasing qualities. Our method draws from the recent success of Adapter (Rebuffi et al., 2017; Houlsby et al., 2019) and Adapter-Fusion (Pfeiffer et al., 2021) networks, originally introduced in the context of multi-task learning to enable parameter-efficient training and to avoid catastrophic forgetting. Specifically in multi-task learning with AdapterFusion, each task is encoded in specific parameter-efficient adapter networks attached to the transformer blocks (Vaswani et al., 2017) of a PLM, and the knowledge across the tasks is shared through a FusionAdapter layer with an attention mechanism. DAM extends this principle idea to bias mitigation by viewing the objective of “removing each protected attribute” as a stand-alone task, learned by an independent debiasing adapter module. These debiasing adapters are then integrated into the core model via a fusion layer to mitigate biases. This modular architecture of DAM enables on-demand debiasing, where by simply removing the debiasing adapters the model would return to its original state. The modularity and parameter efficiency of DAM also enables creating and sharing stand-alone debiasing solutions (in form of debiasing adapters), which can be applied to the underlying models on-demand. So, if one uses task adapter’s output in DAM, they get biased embedding else they can take output of fusion layer for debiased embedding.

We evaluate DAM in extensive experiments covering various bias mitigation applications in three classification tasks: We study occupation prediction from biographies (De-Arteaga et al., 2019), mention prediction (Pardo et al., 2016), and hate speech detection (Founta et al., 2018), involving the protected attributes of gender, age, and dialect-based race. Among these, the mention prediction task provides two protected attributes (gender, age), enabling the study of adapter-based bias mitigation on multiple factors. Our evaluation results show that DAM (1) consistently improves bias mitigation in terms of lower leakage of protected attribute in comparison with fully fine-tuned models, and (2) better avoids catastrophic forgetting (Parisi et al., 2019; Chen et al., 2020) in a multi-attribute debiasing scenario, while (3) maintaining on-par task performance. Our results show the benefits of DAM in supporting fairness, and reveal the challenges in the scenarios inherently bounded by trade-offs.

Our contribution is three-fold:

- Introducing DAM, a novel, parameter-efficient approach for on-demand and modularized bias mitigation.
- Conducting a large set of experiments on three datasets and three classification tasks, showing the on-par or better performance of DAM in comparison with strong baselines.
- Examining the confounding cross-attribute effects in a multi-attribute dataset and demonstrating the benefits of our modularized bias mitigation approach.

The remainder of the paper is organized as follows: Section 2 reviews the relevant literature in adapter networks and bias mitigation domains. In Section 3, we explain the architecture and training strategy of DAM. Section 4 describes the setup of the experiments, whose results are reported and discussed in Section 5. Finally, Sections 6 and 7, respectively, summarize and discuss limitations of our work. Our code and trained resources are available in [https://github.com/CPJKU/ModularizedDebiasing](https://github.com/CPJKU/ModularizedDebiasing).

2 Related Work

2.1 Adapter Networks

Adapter networks have been introduced in the context of multi-task learning (Rebuffi et al., 2017), Houlsby et al. (2019), and later Stickland and Murray (2019) attach the adapter module to the transformer blocks of a PLM, and learn a task by only updating the adapter parameters, while keeping the PLM’s parameters unchanged. These studies show that the highly parameter-efficient adapter approach in general performs on par with fine-tuning
all parameters of a BERT model (Devlin et al., 2019) for many tasks.

Other studies investigate various characteristics of adapter-based models such as the parameter efficiency, architectural variations, and transfer learning across tasks. Rücklé et al. (2021) show the training efficiency of adapter models in comparison with full model finetuning (∼60%). Han et al. (2021a) examine the robustness to initialization seeds and training stability of the adapter approach. Mahabadi et al. (2021b) propose more parameter-efficient models by sharing adapter parameters across layers, followed by Mahabadi et al. (2021a), who introduce even more compact adapter modules. Poth et al. (2021) investigate the potential knowledge transfer across tasks. Recently, Pfeiffer et al. (2021) introduce AdapterFusion, where a fusion layer is defined on top of pre-trained adapter modules, and learns to combine information of adapters via an attention mechanism. This approach, avoiding the common pitfalls of catastrophic forgetting (Parisi et al., 2019), enables an effective knowledge transfer across tasks. Our work contributes to this direction by extending the concept of AdapterFusion to the task of bias mitigation via supervised interaction between the downstream task and the protected attribute.

2.2 Fairness & Bias Mitigation in NLP

The existence of biases and stereotypes in PLMs has been reported and discussed in several studies (Nadeem et al., 2021; Bhardwaj et al., 2021; Liang et al., 2021; Kirk et al., 2021; Vig et al., 2020). PLMs may even exacerbate these biases in downstream tasks as shown e.g. in the context of IR (Rekabsaz and Schedl, 2020). To reduce biases, in-processing methods – the focus of this work – aim at reducing the (co)relation between the model’s internal representations and the protected attributes (Ganhör et al., 2022). Debiasing PLMs has been approached for instance by linearly projecting embeddings to a new space that removes correlations to protected attributes (Kaneko and Bollegala, 2021; Bolukbasi et al., 2016). In a similar spirit, Guo et al. (2022) introduce a distribution alignment loss to force the model’s outputs to become independent of the protected attribute. Schick et al. (2021) recently show that the encoded information in models can be exploited to spot the biases and hence to penalize them.

Adversarial training is a commonly used method to learn representations invariant to a specific factor of variation. Xie et al. (2017) and later Madras et al. (2018) introduce adversarial learning to the context of fair representation learning, where an adversary network learns to predict the protected attribute, and exploits the gradient of this prediction to remove the protected information using a gradient reversal layer (Ganin and Lempitsky, 2015). Several works further investigate the use of adversarial training for removing demographic information from neural/PLM-based text classifiers (Elazar and Goldberg, 2018; Barrett et al., 2019; Wang et al., 2021). Notably, Han et al. (2021b) show the benefit of having an ensemble of orthogonal adversaries. Beyond classification, Rekabsaz et al. (2021a) show that by applying adversarial training in the context of IR, one can achieve a more balanced retrieval of documents with respect to the presence of protected attributes in their contents.

As alternatives to adversarial debiasing, other works approach bias mitigation by minimizing the approximated upper bound of mutual information between task and protected attribute (Cheng et al., 2020b; Colombo et al., 2021), contrastive learning for fair representation disentanglement (Cheng et al., 2020a; Zhang et al., 2021), and introducing list-wise fairness regularization (Zerveas et al., 2022). While the mentioned methods are applied to whole a model, Iterative Nullspace Projection (INLP) (Ravfogel et al., 2020) achieves debiased representations by finding a linear mapping applied to output embeddings. The linear mapping of the INLP method and similarly the one of Ravfogel et al. (2022) offer effective bias mitigation methods particularly designed for the models with a linear decoder, but are not necessarily suited for the cases with non-linear decoders (e.g., a language generation decoder, or non-linear attackers).

Regarding adapter-based debiasing and closely related to our work, Lauscher et al. (2021) recently utilize adapters for debiasing PLMs by training an adapter which removes a protected attribute using counterfactual augmented data. When applied to a downstream task, another adapter is added on top of the debiasing adapter, and trained according to the task’s objective. While shown effective in practice, in this stacking architecture the adapters in the higher levels inherently depend on the ones in the lower levels. They can not be learned stand-alone. In contrast, the fusion-based approach of DAM enables control by learning modular and
independent debiasing adapters, which supports flexible plug-in/plug-out on demand.

3 Bias Mitigation with DAM

We approach the scenario in which the aim is to maximize a task’s core objective (such as occupation prediction or hate speech prediction), while reducing the effects of the biases caused by \( k \) protected attributes. The crux of the proposed DAM approach is to first learn a task adapter and \( k \) bias removal adapters all independently, and then combine them through a fusion module. Once a model has been trained with DAM, one can easily switch between the original model by only using the task adapter, or alternatively impose debiasing by “plugging in” the learned debiasing adapters as well as the fusion module. In what follows, we explain the architecture of DAM followed by the optimization procedure.

3.1 Model Architecture

DAM consists of three main parts: task adapter, \( k \) debiasing adapters, and fusion. Following Pfeiffer et al. (2021), and as shown in Figure 1, these parts extend the architecture of a transformer block by being added to its last layer.

Adapters Each adapter is defined as a multilayer perceptron (one hidden layer and \( \tanh(x) \) in our experiments), where the hidden layer typically has the same or a smaller dimension than the input. In DAM, the task adapter and the \( k \) debiasing adapters receive the output vector of the preceding transformer components, denoted as \( u \), and apply the corresponding transformations to output the vectors \( v_t \), and \( v_{b_1}, \ldots, v_{b_k} \), respectively.

Fusion The fusion module combines the outputs of the task and debiasing adapters through the attention mechanism to produce the final output vector. This module receives \([v_t, v_{b_1}, \ldots, v_{b_k}]\) as keys and values, and the \( u \) vector as the query of a single-head multiplicative attention network, and calculates the output as the weighted sum of the value vectors. These weights are calculated as attention scores and form a probability distribution over value vectors. Essentially, the fusion module learns to perform a linear combination of the embedding containing information for the task with the embeddings of the debiasing adapters, to provide the final, debiased embedding.

3.2 Learning On-demand Bias Mitigation

DAM introduces a generic approach to bias mitigation and representation disentanglement, and can, in principle, be integrated with any bias removal optimization method, provided that the method defines separate losses for the task and each of the protected attributes. Irrespective of the optimization method, the training procedure of DAM is the following: first, the task adapter is trained according to the task’s objective. Then, one debiasing adapter for each protected attribute is trained using only its own debiasing objective (see next paragraph), without involving the task loss. Finally, all adapters’ parameters are kept frozen and the fusion layer is trained using a combination of the task objective and debiasing objectives. Throughout training, the parameters of the underlying PLM remain frozen. In what follows, we describe using DAM together with the adversarial bias mitigation method as depicted in Figure 2.

Adversarial Training Adversarial bias mitigation aims to make the output embedding of a model invariant to the protected attribute, by removing information that allows predicting protected attributes from the latent representation. In this sense, adversarial training follows a fairness through blindness (Barocas et al., 2019) approach, where the system is made agnostic of the variations
in underlying protected attributes.

To apply adversarial debiasing with DAM (see Figure 2) in the context of text classification, the model first encodes the input sequence $x$ in the latent vector $z$, based on which the corresponding class is predicted using a task classification head. The loss of this prediction, denoted as $L_t$, is defined as cross entropy, and its gradient is used to update the parameters of the task adapter. Similarly, a dedicated classification head is defined for each protected attribute $b_i$, which also receives $z$ as input, predicts the corresponding protected attribute, and calculates the cross entropy loss function $L_{b_i}$. A common approach to removing the information of $b_i$ encoded in $z$ is gradient reversal layer (GRL) (Ganin and Lempitsky, 2015) added before the debiasing head. GRL multiplies the gradient of $L_{b_i}$ with a factor of $-\gamma_i$, and thereby simplifies the learning process to a standard gradient-based optimization. In DAM, this reversed gradient of $L_{b_i}$ is used to learn the parameters of the debiasing adapter corresponding to the protected attribute $b_i$. Once adapters are trained, the fusion layer is learned jointly according to all task and debiasing objectives, as formulated below:

$$L_{\text{Fusion}}^{\text{Adv}} = L_t + \sum_{i=1}^{k} L_{b_i}$$  \hspace{1cm} (1)$$

Note that, while no weights are used to directly scale the individual loss functions, the effects of bias mitigation losses on model parameters are adjusted via their corresponding $\gamma_i$ hyperparameters.

4 Experiment Setup

We evaluate DAM on three classification tasks and compare the results with a set of strong baselines.

Tasks and Data We conduct our experiments on three classification tasks. The first task is biography classification using the BIOS (De-Arteaga et al., 2019) dataset. The objective of the task is to predict a person’s job given their biography, where the name and any indication of the person’s gender (such as pronouns) in the biography is omitted. The protected attribute is the person’s gender. The BIOS dataset contains around 430K data points with 28 occupations as the core task’s classes, and two protected attribute classes (female/male).

The second task is hate speech detection, a sensitive task, as its use in any automated system with biases towards any societal group can be extremely delicate (e.g. in automatic social media moderation). Xia et al. (2020) show the existence of a strong correlation between dialect-based racial bias and hate speech detection. In our experiments, we use FDCL18 (Founta et al., 2018), a dataset of tweets annotated with four hate speech labels: normal, spam, abusive, and hateful. Following previous works (Sap et al., 2019; Ravfogel et al., 2020; Zhang et al., 2021), we assign artificial race labels of African American, and White American to FDCL18 (more details in Appendix A.1). In this setting, the main task is hate speech classification, and the protected attribute is the dialect-based racial identity of the tweet author. Our final dataset contains approximately 62K data points.

While the first two tasks contain one protected attribute, in our third classification task, we aim to explore the effect of more than one biases on our models. We therefore opt for a subset of the PAN16 dataset (Pardo et al., 2016). This dataset has been designed for the task of mention detection, namely to predict whether a tweet includes the mention of other users, and the protected attributes are gender and age of the tweet’s author. The dataset contains approximately 200K data points with binary task classes (mention, no mention), as well as two gender labels and five age groups. Further details on the three datasets are provided in Appendix A.1.

Models and Baselines We conduct the experiments on the following models: FT: fully fine-tuning the PLM on the task. FT-DEBIAS: fully fine-tuning the PLM on the task as well as on the corresponding adversarial debiasing objective(s). INLP: using the implementation and suggested hyperparameter setting by Ravfogel et al. (2020), in particular with a linear task classifier (Logistic

<table>
<thead>
<tr>
<th>Model</th>
<th># Params</th>
</tr>
</thead>
<tbody>
<tr>
<td>FT</td>
<td>11,237,380</td>
</tr>
<tr>
<td>FT-DEBIAS</td>
<td>11,568,910</td>
</tr>
<tr>
<td>BERT-Mini</td>
<td></td>
</tr>
<tr>
<td>ADP</td>
<td>330,500</td>
</tr>
<tr>
<td>ADP-DEBIAS</td>
<td>432,206</td>
</tr>
<tr>
<td>DAM</td>
<td>1,186,830</td>
</tr>
</tbody>
</table>

Table 1: Number of trainable parameters. The adapter-based models contain significantly fewer trainable parameters in comparison with fully fine-tuned models.
Table 2: Results on BIOS and FDCL18 datasets for the task classifiers (accuracy) and protected attributes’ attackers (accuracy–Acc and balanced accuracy–BAcc), reported as the mean and standard deviation (in subscript) over three runs. Arrows indicate the direction of better results. On each evaluation metric and each PLM, the best results among adapter-based models are shown in **bold**, and over all models by symbol ♣.

Regressor. **INLP-NONLIN**: the INLP model with the same setting, but instead of Logistic Regressor, INLP-NONLIN uses a non-linear task classifier: a two-layer feed-forward layer with a tanh activation. This setting follows the same configuration of all other baselines/models with respect to the task classifier. **ADP**: adding one adapter to the PLM and training it only through the task objective. **ADP-DEBIAS**: adding one adapter to the PLM and training it through the task objective as well as the adversarial debiasing objective(s). Finally, our proposed **DAM** model, which consists of one task adapter and separate debiasing adapters for each protected attribute, all combined with a fusion module. DAM uses the same task adapter trained by ADP and further learns the debiasing adapters and fusion layer. We should note that, since through adapter-based training the task performance of ADP might be different from the one of FT, a fair comparison of the core task performance should be between the results of ADP and its respective DAM model. We emphasize that we are especially interested in adapters because they preserve the original model and can be disengaged at will, to trade off debiasing for utility/ performance.

As the PLM encoder for all models, we conduct our experiments using two versions of BERT (Devlin et al., 2019) with different sizes, namely BERT-Mini (Turc et al., 2019) and BERT-Base in order to provide a more comprehensive picture regarding the effect of encoder size and number of involved parameters. The number of trainable parameters of the models is reported in Table 1. The complete details of our hyperparameters setting and training procedure are explained in Appendix A.2 and Appendix A.3, respectively.

**Evaluation and Attackers** To evaluate the performance of the classifiers on the core task, we use the accuracy metric. To evaluate bias mitigation, following previous works (Elazar and Goldberg, 2018; Barrett et al., 2019), we report the leakage of a protected attribute in terms of accuracy and balanced accuracy of an attacker. To train an attacker, we freeze the model’s parameters, and train a new classification head (two-layer feed-forward layer with a tanh activation) to predict the protected attribute from the z vector. We train an ensemble of 5 attackers with the same configuration and training procedure, and report the results of the best performing one. We report the performance of the attacker in terms of accuracy and also balanced/macro accuracy (average of per-class ac-
Table 3: Results of the PAN16 mention detection task with gender (G) and age (A) as protected attributes on the task classifiers (accuracy) and protected attributes’ attackers (accuracy–Acc and balanced accuracy–BAcc). The mean and standard deviation (in subscript) over three runs are reported. On each evaluation metric and each PLM, the best results among adapter-based models are shown in bold, and over all models by symbol ♣.

### 5 Results and Discussion

In this section, we first discuss the results on the datasets with one protected attribute (BIOS and FDCL18), followed by investigating the effect of DAM on PAN16 with two protected attributes.

#### Single-attribute bias mitigation

The results on BIOS and FDCL18 datasets are reported in Table 2. We start with the results of BERT-Mini on both datasets. As shown, consistent with previous stud-
ies the task performance of ADP remains on par with the full model fine-tuning FT. In both FT and ADP, we observe that the adversarial debiasing (FT-DEBIAS and ADP-DEBIAS respectively) significantly reduces leakage in terms of Acc and BAcc of attackers, while task performance remains on par with the corresponding baseline models. Overall, adversarial methods outperform baseline INLP models, particularly in respect to maintaining task performance during bias mitigation.

Compared to ADP and ADP-DEBIAS, we observe that DAM improves ADP-DEBIAS in terms of bias mitigation, while maintaining or even slightly improving performance on the task. The difference between DAM and ADP-DEBIAS indicates the benefit of learning separate adapters for performing the task and bias mitigation, and then combining them. Also comparing DAM with FT-DEBIAS, overall DAM performs on par in terms of task performance and leakage, while DAM provides the additional benefits of selectively engaging bias mitigation and significantly fewer trainable parameters.

Comparing the results between the core PLMs, we see that the above observations for BERT-Mini also hold true for BERT-Base on the BIOS dataset. On the FDCL18 dataset with BERT-Mini, DAM is in fact the only bias mitigation approach that can effectively prevent leakage while also preserving performance. It is the best in terms of leakage reduction, while its task performance is better than ADP. The situation differs for results of BERT-Base on FDCL18, as leakage balanced accuracy of FT-DEBIAS is surprisingly low.

Multi/Two-attribute bias mitigation Table 3 reports the results on PAN16 with two protected attributes, gender (G) and age (A). To observe the effect of single- versus multiple-attribute debiasing, we train the debiasing models (FT-DEBIAS, ADP-DEBIAS, and DAM) in three variations denoted with subscripts G, A, and G+A, indicating the debiasing optimization on only gender, only age, and both gender and age, respectively.\(^1\)

Observing similar patterns to the previous experiments, DAM\(_G\), DAM\(_A\), and DAM\(_{G+A}\) provide a significant improvement in bias mitigation (lower leakage) in comparison with their respective debiasing baseline variations.\(^2\) Also similar to single-attribute datasets, ADP performs slightly worse than FT in respect to task performance, but DAM models remain on par with ADP or slightly improve it. We observe these patterns consistently on both BERT-Mini and BERT-Base. Overall, DAM provides equal or better bias mitigation in comparison with FT-DEBIAS and INLP, while offering on-demand debiasing and parameter efficiency.

Particularly with respect to multi-attribute bias mitigation, the results show the benefits of DAM\(_{A+G}\) in separately learning multiple debiasing adapters and combining them, when compared with FT-DEBIAS\(_{G+A}\), which simultaneously learns both debiasing factors. In particular, looking at the attacker’s BAcc, while FT-DEBIAS\(_{G+A}\) is not able to debias the age attribute (in comparison to FT), our DAM\(_{G+A}\) provides strongly improved bias mitigation for age (and also gender), with similar or better results to only optimizing for gender or age (DAM\(_G\) and DAM\(_A\), respectively). This highlights the effectiveness of DAM in preventing catastrophic forgetting of debiasing functionalities in multi-attribute scenarios.\(^3\) We further provide an investigation of the attention distributions of DAM’s fusion module in Appendix B.

6 Conclusion

We propose a novel bias mitigation approach which enables flexible switching between the original and debiased state of a model. Our proposed DAM method extends the idea of multi-task learning using AdapterFusion to bias mitigation, by first learning the main task and the debiasing objectives as separate adapters, and then leveraging the attention-based fusion approach to merge these adapters and deliver debiased results. Our experiments on three classification tasks show that, beside flexible switching, DAM improves bias mitigation, provides on par classification performance to vanilla adversarial debiasing, and addresses the issue of catastrophic forgetting in multi-attribute bias mitigation.

\(^1\)We run INLP and INLP-NONLIN on the two variations of G and A, but we are not aware of a principled way for this method to simultaneously debias more than one protected attribute.

\(^2\)BAcc is more suited for comparisons of age on this dataset, as the age labels are highly imbalanced.

\(^3\)Our experiments using DAM also indicate a possible correlation between Gender and Age, as debiasing Gender (DAM\(_G\)) leads to slight reduction in Age leakage, while similarly, debiasing Age (DAM\(_A\)) results in a slight removal of gender information.
7 Limitations

In the experiments, gender is considered as a binary construct due to practical constraints. In particular, in BIOS and PAN16 the gender is provided only in the form of male and female. We are however fully aware that a gender binary model is not representative of all individuals; yet working with in-the-wild data (as in our case) entails an unavoidable caveat, derived from the still predominant belief that human beings can be sorted into two discrete categories (Hyde et al., 2019). However, the proposed method can still be defined for generic non-binary settings and can be applied to any sensitive attribute with more than two categories, as exemplified by our consideration of age classes.

We should also highlight two limitations of this study in respect to the model and optimization. First, adversarial approaches in general (including our proposed approach) aims to reduce the correlations in the model to the protected attribute based on the observed data. This approach, like other data-oriented bias mitigation methods, might lack effective generalization, particularly when the model is evaluated on other domains or out-of-distribution data. The second limitation regarding the discussed bias mitigation methods based on adversarial training (also involving our proposed approach) is that, while the method aims to make the model prediction agnostic to protected attributes, it does not directly account for a balanced treatment of subpopulations in regards to the utility metrics and quality of the received service.

8 Acknowledgment

This work received financial support by the Austrian Science Fund (FWF): P33526 and DFH-23; and by the State of Upper Austria and the Federal Ministry of Education, Science, and Research, through grants LIT-2020-9-SEE-113 and LIT-2021-YOU-215.

References


Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to computer programmer as woman is to homemaker: debiasing word embeddings. Advances in Neural Information Processing Systems.


Francisco Manuel Rangel Pardo, Paolo Rosso, Ben Verhoeven, Walter Daelemans, Martin Potthast, and Benno Stein. 2016. Overview of the 4th author profiling task at pan 2016: Cross-genre evaluations. In CLEF.


Iulia Ture, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: On the importance of pre-training compact models.


A Experiment Settings – Additional Details

A.1 Datasets

In FDCL18 dataset, we use the TwitterAAE model (Blodgett et al., 2016) to assign racial dialect classes. The TwitterAAE model predicts four racial classes, African American, White American, Hispanic, and Others. We labeled a tweet as African American or White American if the prediction score was greater than 0.5. For PAN16 dataset, following (Sap et al., 2019) we balanced the task labels and sampled 200K data. The age groups of this dataset are 18-24, 25-34, 35-49, 50-64, and 65+. The proportions of data items regarding the labels of the task and protected attributes in the three dataset are as follows:

- BIOS dataset: Task (dentist: 0.03, professor: 0.27, teacher: 0.04, psychologist: 0.04, nurse: 0.07, poet: 0.01, photographer: 0.06, journalist: 0.04, filmmaker: 0.02, physician: 0.08, composer: 0.2, attorney: 0.08, model: 0.03, painter: 0.02, accountant: 0.01, pastor: 0.01, comedian: 0.01, surgeon: 0.05, architect: 0.03, paralegal: 0.01, dj: 0.01, chiropractor: 0.01, software engineer: 0.02, dietitian: 0.02, rapper: 0.01, personal trainer: 0.003, yoga teacher: 0.01, interior designer: 0.01); Gender (Female: 0.5, Male: 0.5)
- FDCL18 dataset: Task (normal: 0.73, spam: 0.12, abusive: 0.12, hateful: 0.04); Race (White: 0.5, AA: 0.5)
- PAN16 dataset: Task (Mention: 0.5, Not Mention: 0.5); Gender (male: 0.54, female: 0.46); Age ( 3: 0.18, 1: 0.34, 2: 0.40, 0: 0.07, 4: 0.01)

A.2 Hyperparameter setting

Across experiments, we keep specific hyperparameters consistent. Batch size is 64, learning rate is 2e-5 (except for training task and debiasing adapter as explained below), training epochs is 20, dropout probability is 0.3, and adversarial debiasing coefficient is 1 for all models (when applicable).

For task adapter and debiasing adapters, we tune the learning rate and the hidden layer dimension of adapter. We conduct brute search over the learning rate values of [1e-5, 1e-4, 1e-3, 1e-2], and hidden layer dimension with a division factor of [1/2, 1/4, 1/8, 1/16]. For INLP we use 10 iterations and 10 classifiers to learn null space while for INLP-NonLIN we same setting (300 iterations and 300 classifiers) as in (Ravfogel et al., 2020).

A.3 Training procedure

We randomly split the dataset into train, validation, and test set with the proportions 63:12:15 for BIOS, 63:12:15 for FDCL18, and 80:5:15 on PAN16. We use the validation set for hyperparameter tuning, and the best result on the validation set is evaluated on test set for the final results. The validation and test sets in all datasets follow the same distribution as the whole dataset. To address the unbalancedness of the dataset and the potential problems in adversarial learning, we apply upsampling only on the training sets of BIOS and FDCL18 datasets, to balance the protected attribute labels within each task label. For instance, genders are balanced in the dentist class by repeating the data items of the minority subgroup.

B Fusion Attention Analysis

We investigate the attention distribution of the fusion network and observe the weights it gives to the adapters. Figures 3a, 3b, and 4 depict the attention scores of each adapter averaged over all fusion layer in DAM for BIOS, and FDCL18, and PAN16 datasets, respectively. To avoid confusion in visualization, we only used 4% of data points randomly sampled from test set. As shown, the task adapter has weights close to 1, while debiasing adapters are assigned attention scores slightly higher than 0. The top three outliers in BIOS with the highest attentions on the debiasing adapter are reported in Table 4.

<table>
<thead>
<tr>
<th>Text</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>passionately promotes healthy dietary and lifestyle choices to prevent disease and achieve optimal health practices include clayton heights family dental street panorama family dental avenue and surrey family dental avenue primary research interests include collective security and global health in an international relations and international political economy perspective</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4
Figure 3: Attention distribution of a set of sampled data points in the fusion module of DAM.

Figure 4: Attention distribution of a set of sampled data points in the fusion module of DAM in PAN16 dataset.