Adversarial Scrubbing of Demographic Information for Text Classification

Somnath Basu Roy Chowdhury **Sayan Ghosh** **Yiyuan Li** **Junier B. Oliva**
{somnath, sayghosh, yiyuanli, joliva}@cs.unc.edu

Shashank Srivastava **Snigdha Chaturvedi**
{ssrivastava, snigdha}@cs.unc.edu
UNC Chapel Hill

Abstract

Contextual representations learned by language models can often encode *undesirable attributes*, like demographic associations of the users, while being trained for an unrelated target task. We aim to scrub such undesirable attributes and learn fair representations while maintaining performance on the target task. In this paper, we present an adversarial learning framework "**Ad**versarial **S**crubber" (ADS), to debias contextual representations. We perform theoretical analysis to show that our framework converges without leaking demographic information under certain conditions. We extend previous evaluation techniques by evaluating debiasing performance using Minimum Description Length (MDL) probing. Experimental evaluations on 8 datasets show that ADS generates representations with minimal information about demographic attributes while being maximally informative about the target task.

1 Introduction

Automated systems are increasingly being used for real-world applications like filtering college applications (Basu et al., 2019), determining credit eligibility (Ghailan et al., 2016), making hiring decisions (Chalfin et al., 2016), etc. For such tasks, predictive models are trained on data coming from human decisions, which are often biased against certain demographic groups (Mehrabi et al., 2019; Blodgett et al., 2020; Shah et al., 2020). Biased decisions based on demographic attributes can have lasting economic, social and cultural consequences.

Natural language text is highly indicative of demographic attributes of the author (Koppel et al., 2002; Burger et al., 2011; Nguyen et al., 2013; Verhoeven and Daelemans, 2014; Weren et al., 2014; Rangel et al., 2016; Verhoeven et al., 2016; Blodgett et al., 2016). Language models can often encode such demographic associations even without having direct access to them. Prior works have shown that intermediate representations in a deep learning model encode demographic associations of the author or person being spoken about (Blodgett et al., 2016; Elazar and Goldberg, 2018; Elazar et al., 2021). Therefore, it is important to ensure that decision functions do not make predictions based on such representations.

In this work, we focus on removing demographic attributes encoded in data representations during training text classification systems. To this end, we present "**Ad**versarial **S**crubber" (ADS) to remove information pertaining to *protected attributes* (like gender or race) from intermediate representations during training for a *target task* (like hate speech detection). Removal of such features ensures that any prediction model built on top of those representations will be agnostic to demographic information during decision-making.

ADS can be used as a plug-and-play module during training any text classification model to learn fair intermediate representations. The framework consists of 4 modules: Encoder, Scrubber, Bias discriminator and Target classifier. The Encoder generates contextual representation of an input text. Taking these encoded contextual representations as input, the Scrubber tries to produce fair representations for the target task. The Bias discriminator and Target classifier predict the protected attribute and target label respectively from the Scrubber's output. The framework is trained end-to-end in an adversarial manner (Goodfellow et al., 2014).

We provide theoretical analysis to show that under certain conditions Encoder and Scrubber converge without leaking information about the protected attribute. We evaluate our framework on 5 dialogue datasets, 2 Twitter-based datasets and a Biographies dataset with different target task and protected attribute settings. We extend previous evaluation methodology for debiasing by measuring Minimum Description Length (MDL) (Voita and Titov, 2020) of labels given representations,

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 550–562
November 7–11, 2021.

instead of probing accuracy. MDL provides a finer-grained evaluation benchmark for measuring debiasing performance. We compute MDL using off-the-shelf classifiers[1] making it easier to reproduce. Upon training using ADS framework, we observe a significant gain in MDL for protected attribute prediction as compared to fine-tuning for the target task. Our contributions are:

- We present **Ad**versarial **S**crubber (ADS), an adversarial framework to learn fair representations for text classification.
- We provide theoretical guarantees to show that Scrubber and Encoder converge without leaking demographic information.
- We extend previous evaluation methodology for adversarial debiasing by framing performance in terms of MDL.
- Experimental evaluations on 8 datasets show that models trained using ADS generate representations where probing networks achieve near random performance on protected attribute inference while performing similar to the baselines on target task.
- We show that ADS is scalable and can be used to remove multiple protected attributes simultaneously.

2 Related Work

Contextual representations learned during training for a target task can be indicative of features unrelated to the task. Such representations can often encode undesirable demographic attributes, as observed in unsupervised word embeddings (Bolukbasi et al., 2016) and sentence embeddings (May et al., 2019). Prior work has analysed bias in different NLP systems like machine translation (Park et al., 2018; Stanovsky et al., 2019; Font and Costa-Jussa, 2019; Saunders and Byrne, 2020), NLI (Rudinger et al., 2017), text classification (Dixon et al., 2018; Kiritchenko and Mohammad, 2018; Sap et al., 2019; Liu et al., 2021), language generation (Sheng et al., 2019) among others.

Debiasing sensitive attributes for fair classification was introduced as an optimization problem by Zemel et al. (2013). Since then, adversarial training (Goodfellow et al., 2014) frameworks have been explored for protecting sensitive attributes for NLP tasks (Zhang et al., 2018; Li et al., 2018; Elazar and Goldberg, 2018; Liu et al., 2020).

[1]We use MLPClassifier modules from scikit-learn.

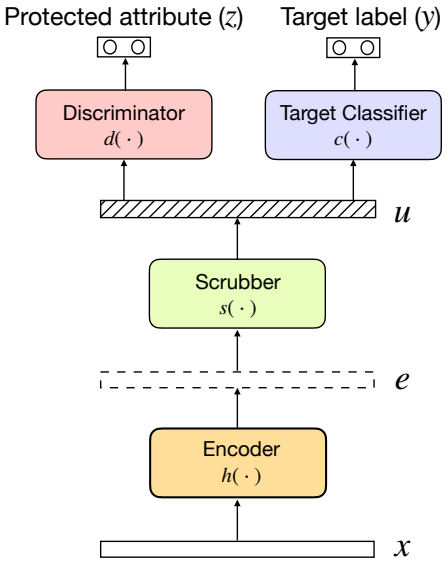


Figure 1: Architecture of the Adversarial Scrubber (ADS). Encoder receives an input x to produce e. Scrubber uses e to produce u. Bias discriminator d and Target classifier c infer protected attribute z and target task label y from u.

Our work is most similar to Elazar and Goldberg (2018), which achieves *fairness by blindness* by learning intermediate representations which are oblivious to a protected attribute. We compare the performance of ADS with Elazar and Goldberg (2018) in our experiments.

3 Adversarial Scrubber

ADS takes text documents $\{x_1, x_2, \ldots, x_n\}$ as input from a dataset $\mathcal{D}$ with corresponding target labels $\{y_1, y_2, \ldots, y_n\}$. Every input x_i is also associated with a *protected attribute* $z_i \in \{1, 2, ...K\}$. Our goal is to construct a model $f(x)$ such that it doesn't rely on z_i while making the prediction $y_i = f(x_i)$. The framework consists of 4 modules: (i) Encoder $h(\cdot)$ with weights θ_h, (ii) Scrubber $s(\cdot)$ with weights θ_s, (iii) Bias discriminator $d(\cdot)$ with weights θ_d and (iv) Target classifier $c(\cdot)$ with weights θ_c as shown in Figure 1. The Encoder receives a text input x_i, and produces an embedding $e_i = h(x_i)$, which is forwarded to the Scrubber. The goal of the Scrubber is to produce representation $u_i = s(h(x_i))$, such that y_i can be easily inferred from u_i by the Target classifier, c, but u_i does not have the information required to predict the protected attribute z_i by the Bias discriminator d. Our setup also includes a Probing network q, which helps in evaluating the fairness of the learned representations.

Algorithm 1 ADS Training algorithm

1: **for** number of training iterations **do**

2: Sample a minibatch $\{x_i, y_i, z_i\}_{i=1}^{m} \sim \mathcal{D}$

3: Bias discriminator d is updated using the gradients:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \mathcal{L}_d(d(u_i), z_i) \tag{1}$$

4: Update the Encoder h, Scrubber s, and Task Classifier c using the gradients:

$$\nabla_{\theta_c, \theta_s, \theta_h} \frac{1}{m} \sum_{i=1}^{m} \Big[\mathcal{L}_c(c(u_i), y_i) - \lambda_1 H(d(u_i)) + \lambda_2 \delta(d(u_i)) \Big] \tag{2}$$

In the rest of this section, we describe ADS assuming a single Bias discriminator. However, ADS can easily be extended to incorporate multiple discriminators for removing several protected attributes (discussed in Section 6.1).

Scrubber: The Scrubber receives the input representation $h(x_i)$ from Encoder and generates representation $u_i = s(h(x_i))$. The goal of the Scrubber is to produce representations such that the Bias discriminator finds it difficult to predict the protected attribute z_i. To this end, we consider two loss functions:

Entropy loss: In the Entropy loss, the Encoder and Scrubber parameters are jointly optimized to *increase* the entropy of the prediction probability distribution, $H(d(u_i))$.

δ-loss: The δ-loss function penalizes the model if the discriminator assigns a high probability to the correct protected-attribute class. For every input instance, we form an output mask $m_i \in \mathbb{R}^{1 \times K}$ where K is the number of *protected attribute classes*. $m_i^{(k)} = 1$ if $z_i = k$ and 0 otherwise. The Encoder and Scrubber *minimizes* the δ-loss defined as:

$$\delta(d(u_i)) = m_i^T \text{softmax}_{gumble}(d(u_i)) \tag{3}$$

where $\text{softmax}_{gumble}(\cdot)$ is the gumble softmax function (Jang et al., 2017). In our experiments, we use a combination of the entropy and δ losses.

Target classifier: The Target classifier predicts the target label y_i from u_i by optimizing the cross entropy loss: $\mathcal{L}_c(c(u_i), y_i)$.

The Scrubber, Target classifier, and Encoder parameters are updated simultaneously to minimize the following loss:

$$\mathcal{L}_s(e_i, y_i) = \mathcal{L}_c(c(u_i), y_i) - \lambda_1 H(d(u_i)) + \lambda_2 \delta(d(u_i)) \tag{4}$$

where λ_1 and λ_2 are positive hyperparameters.

Bias discriminator: The Bias discriminator, which predicts the protected attribute z_i, is trained to reduce the cross-entropy loss for predicting z_i denoted as $\mathcal{L}_d(d(u_i), z_i)$. The discriminator output is $d(u_i) \in \mathbb{R}^K$, where K is the number of protected attribute classes.

Training: The Bias discriminator and Scrubber (along with Target classifier and Encoder) are trained in an iterative manner as shown in Algorithm 1. First, the Bias discriminator is updated using gradients from the loss in Equation 1. Then, the Encoder, Scrubber and Target classifier are updated simultaneously using the gradients shown in Equation 2.

Probing Network: Elazar and Goldberg (2018) showed that in an adversarial setup even when the discriminator achieves random performance for predicting z, it is still possible to retrieve z using a separately trained classifier. Therefore, to evaluate the amount of information related to y and z present in representations u, we use a probing network q. After ADS is trained, we train q on representations $h(x)$ and $s(h(x))$, to predict y and z (q is trained to predict y and z separately). We consider an *information leak* from a representation, if z can be predicted from it with above random performance. If the prediction performance of q for z is significantly above the random baseline, it means that there is information leakage of the protected attribute and it is not successfully guarded.

4 Theoretical Analysis

Proposition 1. *Minimizing $\mathcal{L}_s$ is equivalent to increasing Bias discriminator loss $\mathcal{L}_d$.*

Proof: Entropy and δ-loss components of $\mathcal{L}_s$ tries to increase the bias discriminator loss. The discriminator cross-entropy loss $\mathcal{L}_d$ can be written as:

$$\begin{aligned}\mathcal{L}_d(v_i, o_i) &= H(v_i, o_i) \\ &= D_{KL}(v_i, o_i) + H(v_i)\end{aligned} \tag{5}$$

where $o_i = d(u_i)$, the Bias discriminator output probability distribution and v_i is a one-hot target distribution $\{v_i \in \mathbb{R}^K, v_i^k = 1 | z_i = k\}$. As $H(o_i)$ increases (Equation 4), $D_{KL}(o_i, v_i)$ value also increases (since v_i is a one-hot vector), thereby increasing $\mathcal{L}_d(o_i, v_i)$ (in Equation 5). Therefore, $\mathcal{L}_d$ increases as we minimize the Scrubber loss component $-H(o_i)$.

The same holds true for the δ-loss component. $\delta(o_i)$ reduces the probability assigned to the true output class which increases the cross entropy loss $\mathcal{L}_d$ (detailed proof provided in Appendix A.2 due to space constraint). Minimizing the entropy and δ-loss components of the Scrubber loss $\mathcal{L}_s$ increases $\mathcal{L}_d$ for a fixed Bias discriminator. Therefore, assuming our framework converges to $(\theta_s^*, \theta_h^*, \theta_d^*)$ using gradient updates from $\mathcal{L}_s$ we have:

$$\mathcal{L}_d(\theta_s^*, \theta_h^*, \theta_d^*) \geq \mathcal{L}_d(\theta_s, \theta_h, \theta_d^*) \tag{6}$$

where (θ_s, θ_h) can be any Scrubber and Encoder parameter setting.

Proposition 2. *Let the discriminator loss $\mathcal{L}_d$ be convex in θ_d, and continuous differentiable for all θ_d. Let us assume the following:*
(a) $\theta_h^{(0)}$ and $\theta_s^{(0)}$ are Encoder and Scrubber parameters when the Scrubber output representation $s(h(x))$ does not have any information about z (one trivial case would be when $s(h(x)) = \vec{0}$, if $\theta_s = \vec{0} \vee \theta_h = \vec{0}$).
(b) $\theta_d^{(0)}$ minimizes $\mathcal{L}_d$ when $s(h(x))$ does not have any information about z (this is achieved when $d(\cdot)$ always predicts the majority baseline for z). $\forall(\theta_s, \theta_h)$, the following holds true:

$$\mathcal{L}_d(\theta_s, \theta_h, \theta_d^{(0)}) = \mathcal{L}_d(\theta_s^{(0)}, \theta_h^{(0)}, \theta_d^{(0)})$$

(c) the adversarial framework converges with parameters θ_s^*, θ_h^* and θ_d^*.
Then, $\mathcal{L}_d(\theta_s^*, \theta_h^*, \theta_d^*) = \mathcal{L}_d(\theta_s^*, \theta_h^*, \theta_d^{(0)})$ which implies that the Bias discriminator loss does not benefit from updates of θ_s and θ_h.

Proof: As the Bias discriminator converges to θ_d^*, we have:

$$\mathcal{L}_d(\theta_s^*, \theta_h^*, \theta_d^*) \leq \mathcal{L}_d(\theta_s^*, \theta_h^*, \theta_d^{(0)}) \tag{7}$$

θ_h and θ_s are updated using gradients from $\mathcal{L}_s$ (Equation 4). Since the Encoder and the Scrubber parameters converge to θ_h^* and θ_s^* respectively, from Proposition 1 (Equation 6) we have:

$$\mathcal{L}_d(\theta_s^*, \theta_h^*, \theta_d^*) \geq \mathcal{L}_d(\theta_s^{(0)}, \theta_h^{(0)}, \theta_d^*) \tag{8}$$

We can show that:

$$\begin{aligned}&\mathcal{L}_d(\theta_s^*, \theta_h^*, \theta_d^{(0)}) \\ &\quad\geq \mathcal{L}_d(\theta_s^*, \theta_h^*, \theta_d^*) && \text{(Equation 7)} \\ &\quad\geq \mathcal{L}_d(\theta_s^{(0)}, \theta_h^{(0)}, \theta_d^*) && \text{(Equation 8)} \\ &\quad\geq \mathcal{L}_d(\theta_s^{(0)}, \theta_h^{(0)}, \theta_d^{(0)}) && \text{(Assumption 2b)} \\ &\quad= \mathcal{L}_d(\theta_s^*, \theta_h^*, \theta_d^{(0)}) && \text{(Assumption 2b)}\end{aligned} \tag{9}$$

Therefore, $\mathcal{L}_d(\theta_s^*, \theta_h^*, \theta_d^*) = \mathcal{L}_d(\theta_s^*, \theta_h^*, \theta_d^{(0)})$.

Proposition 3. *Let us assume that the Bias discriminator $d(\cdot)$ is strong enough to achieve optimal accuracy of predicting z from $s(h(x))$ and assumptions in Proposition 2 hold true. Then, Encoder and Scrubber converge to (θ_h^*, θ_s^*) without leaking information about the protected attribute z.*

Proof: An optimal Bias discriminator $d(\cdot)$ minimizes the prediction entropy, thereby increasing the entropy and δ-loss. Given $(\theta_h^{(0)}, \theta_s^{(0)})$, the Scrubber loss $\mathcal{L}_s$ is maximized for an optimal $\theta_d^{(0)}$ (From Proposition 1, $\mathcal{L}_s(\theta_s^{(0)}, \theta_h^{(0)}, \theta_d^{(0)}) \geq \mathcal{L}_s(\theta_s^{(0)}, \theta_h^{(0)}, \theta_d)$, since $\mathcal{L}_d$ is decreasing with $\delta(o_i)$ and $-H(o_i)$). Then, for any other discriminator θ_d^* we have:

$$\mathcal{L}_s(\theta_s^{(0)}, \theta_h^{(0)}, \theta_d^*) \leq \mathcal{L}_s(\theta_s^{(0)}, \theta_h^{(0)}, \theta_d^{(0)}) \tag{10}$$

Following assumption 2b,do where $\theta_d^{(0)}$ is the optimal Bias discriminator we can show that:

$$\begin{aligned}&\mathcal{L}_s(\theta_s^{(0)}, \theta_h^{(0)}, \theta_d^{(0)}) \\ &\quad\geq \mathcal{L}_s(\theta_s^{(0)}, \theta_h^{(0)}, \theta_d^*) && \text{(Equation 10)} \\ &\quad\geq \mathcal{L}_s(\theta_s^*, \theta_h^*, \theta_d^*) && (\mathcal{L}_s \text{ converges})\end{aligned} \tag{11}$$

Therefore, $\mathcal{L}_s(\theta_s^*, \theta_h^*, \theta_d^*) \leq \mathcal{L}_s(\theta_s^{(0)}, \theta_h^{(0)}, \theta_d^{(0)})$.
From $(\theta_s^{(0)}, \theta_h^{(0)}, \theta_d^{(0)})$, our framework converges to $(\theta_s^*, \theta_h^*, \theta_d^*)$ as the Scrubber loss $\mathcal{L}_s$ decreases

DATASET	Split		
	Train	Dev	Test
Funpedia	24K	2.9K	2.9K
Wizard	3.5K	0.1K	0.1K
ConvAI2	69K	4.5K	4.5K
LIGHT	38K	2.2K	4.5K
OpenSub	210K	25K	29K
DIAL	166K	-	151K
PAN16	160K	-	9K
PAN16	160K	-	10K
Biographies	257K	40K	99K

Table 1: Dataset statistics.

DATASET	z	y	Epoch	λ_1	λ_2
Funpedia	Gender (3)	Sentiment (3)	2	1	1
Wizard	Gender (2)	Sentiment (3)	3	1	0
ConvAI2	Gender (2)	Sentiment (3)	1	1	0
LIGHT	Gender (2)	Sentiment (3)	2	1	0
OpenSub	Gender (2)	Sentiment (3)	2	1	0
DIAL	Race (2)	Sentiment (2)	8	10	0
PAN16	Gender (2)	Mention (2)	5	10	0
PAN16	Age (2)	Mention (2)	3	10	0
Biographies	Gender (2)	Occupation (28)	2	10	0

Table 2: Hyperparameter settings. Each entry for z/y are shown the format "Attribute Name (c)", where c is the number of classes for that attribute.

(Equation 11). Then, from Proposition 2 we have

$$\mathcal{L}_d(\theta_s^*, \theta_h^*, \theta_d^*) \geq \mathcal{L}_d(\theta_s^{(0)}, \theta_h^{(0)}, \theta_d^{(0)})$$

As $\mathcal{L}_d$ does not decrease, and $d(\cdot)$ is optimal it shows that no additional information about z is revealed which the Bias discriminator can leverage to reduce $\mathcal{L}_d$. This shows that starting from $(\theta_s^{(0)}, \theta_h^{(0)}, \theta_d^{(0)})$ where assumptions in Proposition 2 hold, our framework converges to $(\theta_s^*, \theta_h^*, \theta_d^*)$ without revealing information about z.

5 Experiments

In this section, we describe our experimental setup and evaluate ADS on several benchmark datasets.

5.1 Dataset

We evaluate ADS on 5 dialogue datasets, 2 Twitter-based datasets and a Biographies dataset.

(a) **Multi-dimensional bias in dialogue systems**: We evaluate ADS on 5 dialogue datasets: Funpedia, ConvAI2, Wizard, LIGHT and OpenSub, introduced by Dinan et al. (2020). These datasets are annotated with multi-dimensional gender labels: the gender of the person being spoken about, the gender of the person being spoken to, and gender of the speaker. We consider the *gender* of the person being *spoken about* as our protected attribute. The target task in our setup is *sentiment classification*. For obtaining the target label, we label all instances using the rule-based sentiment classifier VADER (Hutto and Gilbert, 2014), into three classes: positive, negative and neutral. The dialogue datasets: Funpedia, Wizard, ConvAI2, LIGHT and OpenSub were downloaded from "md_gender" dataset in huggingface library.[2] We use the same data split provided in huggingface for these dataset.

(b) **Tweet classification**: We experiment on two Twitter datasets. First, we consider the DIAL dataset (Blodgett et al., 2016), where each tweet is annotated with "*race*" information of the author, which is our protected attribute and the target task is *sentiment classification*. We consider two race categories: non-Hispanic blacks and whites. Second, we consider the PAN16 (Rangel et al., 2016) dataset where each tweet is annotated with the author's *age* and *gender* information both of which are protected attributes. The target task is *mention detection*. We use the implementation[3] of Elazar and Goldberg (2018) to annotate both datasets.

(c) **Biography classification**: We evaluate ADS on biographies dataset (De-Arteaga et al., 2019). The target task involves *classification of biographies* into 28 different profession categories, and protected attribute is the *gender* of the person. The dataset has been downloaded and processed from this open-sourced project.[4] We use the same train-dev-test split of 65:10:25 as the authors.

All datasets used in our experiments are balanced. The dataset statistics are reported in Table 1.

5.2 Implementation details

We use a 2-layer feed-forward neural network with ReLU non-linearity as our Scrubber network s. We use BERT-base (Devlin et al., 2019) as our Encoder h. Bias discriminator d and Target classifier c take the pooled output of BERT `[CLS]` representation followed by a single-layer neural network. All the models were using AdamW optimizer with a learning rate of 2×10^{-5}. Hyperparameter details for different datasets are mentioned in Table 2. z and y sections in the table report the protected attribute and the target task for each dataset. For each

[2] https://huggingface.co/datasets/md_gender_bias

[3] https://github.com/yanaiela/demog-text-removal

[4] https://github.com/Microsoft/biosbias

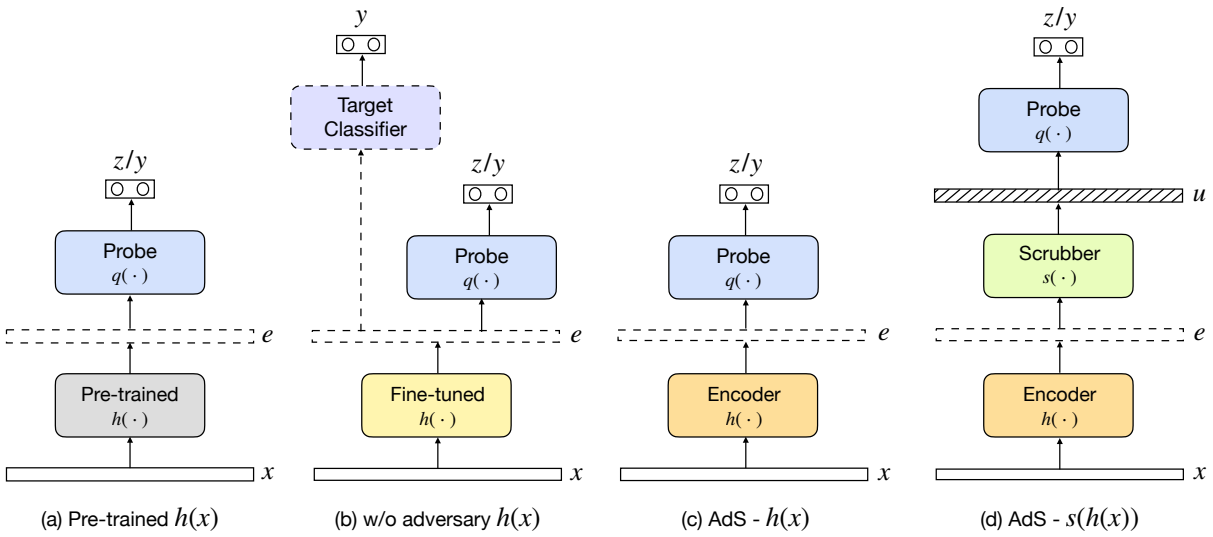


Figure 2: Evaluation setup. We evaluate the performance of the probing network on 4 different representations. (a) Pre-trained $h(x)$ obtained using pre-trained Encoder (b) w/o adversary $h(x)$ when the Encoder h was fine-tuned on the target task (c) ADS $h(x)$ Encoder embeddings and (d) ADS - $s(h(x))$ embeddings from the Scrubber are representations obtained from ADS.

task we also report the number of output classes in paranthesis (e.g. Sentiment (3)). The implementation of this project is publicly available here: https://github.com/brcsomnath/AdS.

5.3 Evaluation Framework

In our experiments, we compare representations obtained from 4 different settings as shown in Figure 2. Figure 2(a), (b) and (c) are our baselines. In Figure 2(a), we retrieve $h(x)$ from pre-trained BERT model. In Figure 2(b), we retrieve $h(x)$ from BERT fine-tuned on the target task. In Figure 2(c), Encoder output $h(x)$ from ADS is evaluated. In Figure 2(d), Scrubber output, $s(h(x))$ is evaluated. This represents our final setup ADS - $s(h(x))$.

5.4 Metrics

We report the F1-score (F1) of the probing network for each evaluation. However, previous work has shown that probing accuracy is not a reliable metric to evaluate the degree of information related to an attribute encoded in representations (Hewitt and Liang, 2019). Therefore, we also report Minimum Description Length (MDL) (Voita and Titov, 2020) of labels given representations. MDL captures the amount of effort required by a *probing network* to achieve a certain accuracy. Therefore, it provides a finer-grained evaluation benchmark which can even differentiate between probing models with comparable accuracies. We compute the online code (Rissanen, 1984) for MDL. In the online setting, blocks of labels are encoded by a probabilistic model iteratively trained on incremental blocks of data (further details about MDL is provided in Appendix A.1). We compute MDL using sklearn's MLPClassifier[5] at timesteps corresponding to 0.1%, 0.2%, 0.4%, 0.8%, 1.6%, 3.2%, 6.25%, 12.5%, 25%, 50% and 100% of each dataset as suggested by Voita and Titov (2020). A higher MDL signifies that more effort is required to achieve the probing performance. Hence, we expect the debiased representations to have higher MDL for predicting z and a lower MDL for predicting y.

6 Results

The evaluation results for all datasets are reported in Table 3. For all datasets, we report performances in 4 settings described in Section 5.3.

Dialogue and Biographies dataset: First, we focus on the results on the dialogue and biographies datasets reported in Table 3 (first two rows). We observe the following: (i) for pre-trained $h(x)$, MDL of predicting z is lower than y for these datasets. This means that information regarding z is better encoded in the pre-trained $h(x)$, than the target label y. (ii) In "w/o adversary $h(x)$" setup, the Encoder is fine-tuned on the target task (without debiasing), upon which MDL for y reduces significantly (lowest MDL achieved in this setting for all datasets) accompanied by a rise in MDL for z. However, it is still possible to predict z with a

[5]We use default hyperparameters from scikit-learn

Dataset → / Setup ↓	FUNPEDIA				WIZARD				CONVAI2			
	Gender (z)		Sentiment (y)		Gender (z)		Sentiment (y)		Gender (z)		Sentiment (y)	
	F1 ↓	MDL ↑	F1 ↑	MDL↓	F1 ↓	MDL ↑	F1 ↑	MDL ↓	F1 ↓	MDL ↑	F1 ↑	MDL ↓
Random	33.3	-	33.3	-	50.0	-	33.3	-	50.0	-	33.3	-
Pre-trained $h(x)$	56.8	24.7	62.3	46.3	78.6	3.8	46.5	7.6	80.3	100.6	62.7	133.7
w/o adversary $h(x)$	51.0	30.9	**92.8**	**2.8**	67.4	5.2	85.1	0.2	72.8	109.0	95.6	**6.5**
ADS - $h(x)$	44.1	35.4	90.3	10.3	63.4	6.5	88.1	0.3	58.3	**134.0**	95.3	10.9
ADS - $s(h(x))$	**29.8**	**41.4**	90.2	10.8	**54.7**	**6.9**	**93.2**	0.2	**56.0**	133.5	95.3	11.0

Dataset → / Setup ↓	LIGHT				OPENSUB				BIOGRAPHIES			
	Gender (z)		Sentiment (y)		Gender (z)		Sentiment (y)		Gender (z)		Occupation (y)	
	F1 ↓	MDL ↑	F1 ↑	MDL ↓	F1 ↓	MDL ↑	F1 ↑	MDL ↓	F1 ↓	MDL ↑	F1 ↑	MDL ↓
Random	50.0	-	33.3	-	50.0	-	33.3	-	50.0	-	3.6	-
Pre-trained $h(x)$	78.6	47.1	60.5	88.7	72.3	192.4	63.9	426.2	99.2	27.6	74.3	499.9
w/o adversary $h(x)$	75.3	55.9	91.4	**8.2**	70.2	311.9	**97.5**	**25.1**	62.3	448.9	99.9	**2.2**
ADS - $h(x)$	60.4	73.8	92.2	16.7	40.7	371.9	96.9	37.4	62.1	444.7	99.9	3.0
ADS - $s(h(x))$	**52.8**	**74.7**	92.3	16.4	40.7	**373.7**	96.9	37.1	**57.1**	**449.5**	99.9	3.3

Dataset → / Setup ↓	DIAL				PAN16							
	Race (z)		Sentiment (y)		Gender (z)		Mention (y)		Age (z)		Mention (y)	
	F1 ↓	MDL ↑	F1 ↑	MDL ↓	F1 ↓	MDL ↑	F1 ↑	MDL ↓	F1 ↓	MDL ↑	F1 ↑	MDL ↓
Random	50.0	-	50.0	-	50.0	-	50.0	-	50.0	-	50.0	-
Pre-trained $h(x)$	74.3	242.6	63.9	300.7	60.9	300.5	72.3	259.7	57.7	302.0	72.8	262.6
w/o adversary $h(x)$	81.7	176.2	**76.9**	99.0	68.6	267.6	89.7	**4.0**	59.0	295.4	89.3	**4.8**
ADS - $h(x)$	69.7	273.0	72.4	51.0	62.3	304.2	89.7	7.1	62.4	302.8	89.3	5.3
ADS - $s(h(x))$	**58.2**	**290.6**	72.9	**56.9**	**48.6**	**313.9**	89.7	7.6	**50.5**	**315.1**	89.2	6.0

Table 3: Evaluation results for all datasets. Expected trends for a metric are shown in ↑- higher scores and ↓- lower scores. Statistically significant best probing performances for z (lowest F1/highest MDL) and y (highest F1/lowest MDL) are in bold.[6] ADS - $s(h(x))$ performs the best in guarding information leak of z for all datasets.

SETUP	DIAL		PAN16			
	Race (z)		Gender (z)		Age (z)	
	Δz	Acc_y	Δz	Acc_y	Δz	Acc_y
w/o adversary LSTM	14.5	67.4	10.1	77.5	9.4	74.7
Elazar and Goldberg (2018)	**4.8**	63.8	4.1	74.3	5.7	70.1
w/o adversary BERT	31.2	**76.4**	18.5	89.7	10.1	**89.3**
ADS - $s(h(x))$	8.2	72.9	**0.8**	**89.8**	**4.7**	89.2

Table 4: Comparing ADS with existing baseline. The best and second best performances are in bold and underlined respectively. ADS - $s(h(x))$ achieve the best performance on both settings in the PAN16 dataset and is able to reduce Δ_z better than baseline on DIAL.

F1-score significantly above the random baseline, (iii) "ADS - $h(x)$" setup achieves similar F1 score for predicting y, but still has a F1-score for z significantly above the random baseline. (iv) "ADS - $s(h(x))$" performs the best in terms of guarding the protected attribute z (lowest prediction F1-score and highest MDL) by achieving near random F1-score across all datasets. It is also able to maintain performance on the target task, as we observe only a slight drop compared to the fine-tuning performance ("w/o adversary $h(x)$" for predicting y).

DIAL & PAN16: Next, we focus on the Twitter-based datasets DIAL & PAN16, where the target task is sentiment classification/mention detection and the protected attribute is one of the demographic associations (race/gender/age) of the author. The evaluation results are reported in Table 3 (third row). For these datasets, we observe that (i) "w/o adversary $h(x)$" representations have higher F1 and lower MDL for predicting z, compared to "Pre-trained $h(x)$". This shows that fine-tuning on the target task y encodes information about the protected attribute z. (ii) "ADS - $h(x)$" performs similar to "w/o adversary $h(x)$" representations on the target task but still leaks significant information about z, unlike the previous datasets. (iii) "ADS - $s(h(x))$" achieves the best performance in terms of guarding the protected variable z (achieves almost random performance in PAN16 dataset), without much performance drop in the target task.

Comparison with Prior Work: We report two metrics following Elazar and Goldberg (2018): (i) Δ_z - which denotes the performance above the random baseline for z (50% for both PAN16 and DIAL) (ii) Acc_y - is the probing accuracy on the

Setup	PAN16					
	Age (z_1)		Gender (z_2)		Mention (y)	
	F1↓	MDL↑	F1↓	MDL↑	F1↑	MDL↓
Random	50.0	-	50.0	-	50.0	-
w/o adversary $h(x)$	66.5	196.4	69.3	192.0	88.6	6.8
ADS $s(h(x))$ - (age)	61.5	224.2	62.6	218.7	88.7	14.3
ADS $s(h(x))$ - (gender)	60.6	222.6	64.2	216.8	88.6	12.9
ADS $s(h(x))$ - (both)	**53.8**	**231.5**	**54.4**	**230.9**	88.6	**5.5**

Table 5: Evaluation results of protecting multiple attributes using ADS. Statistically significant best performances are in bold. Expected trends for a metric are shown in ↑- higher scores and ↓- lower scores. "ADS $s(h(x))$ - (both)" achieves the best performance.[7]

Scrubber loss	Gender (z)			Sentiment (y)		
	F1↓	P↓	R↓	F1↑	P↑	R↑
Random	33.3	33.3	33.3	33.3	33.3	33.3
δ-loss (w/o entropy)	49.5	47.7	53.9	91.2	91.2	91.2
Entropy (w/o δ-loss)	35.7	36.4	53.2	**91.5**	**91.6**	**91.5**
Entropy + δ-loss	**29.8**	**33.3**	**27.0**	90.2	90.5	89.9

Table 6: Ablation experiments on Funpedia using F1-score (F1), Precision (P) and Recall (R). Expected trends for a metric are shown in ↑- higher scores and ↓- lower scores. ADS with both loss components performs the best in guarding z.

target task. Our framework cannot be directly compared with Elazar and Goldberg (2018) as they have used LSTM Encoder. Therefore, we report the baseline Encoder performances as well. In Table 4, we observe that it possible to retrieve z and y from "w/o adversary BERT" with a higher performance compared to "w/o adversary LSTM". This indicates that BERT encodes more information pertaining to both y and z compared to LSTM. In the DIAL dataset, ADS is able to reduce Δ_z by an absolute margin of 25% compared to 9.7% by Elazar and Goldberg (2018), while the absolute drop in Acc_y is 3.5% compared to 3.6% by Elazar and Goldberg (2018). In PAN16 dataset, ADS achieves the best Δ_z and Acc_y performance for both setups with protected attributes: age and gender respectively. ADS - $s(h(x))$ also achieves performance comparable to the "w/o adversary BERT" setup, which is fine-tuned on the target task. Therefore, ADS is successful in scrubbing information about z from the representations of a stronger encoder compared to Elazar and Goldberg (2018).

6.1 Scrubbing multiple protected attributes

In this experiment, we show that using ADS it is possible to guard information about *multiple* protected attributes. $\mathcal{L}_s$ in this setup is defined as:

$$\mathcal{L}_s(e_i, y_i) = \mathcal{L}_c(c(u_i), y_i) - \lambda_1 \sum_{n=1}^{N} H(d_n(u_i)) + \lambda_2 \sum_{n=1}^{N} \delta(d_n(u_i))$$

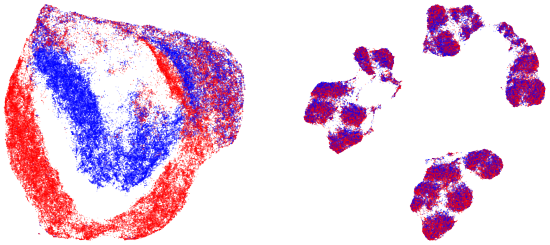
where N is the number of protected attributes and $d_n(\cdot)$ is the Bias discriminator corresponding to the n^{th} protected attribute z_n.

We evaluate on PAN16 dataset considering two protected attributes z_1 (age) and z_2 (gender). The target task is mention prediction. We consider the subset of PAN16 that contains samples with both gender and age labels. This subset has 120K training instances and 30K test instances. Evaluation results are reported in Table 5. Similar to previous experiments, we observe that "w/o adversary $h(x)$" (fine-tuned BERT) leaks information about both protected attributes age and gender. We evaluate the information leak when "ADS $s(h(x))$" is retrieved from a setup with single Bias discriminator (age/gender). We observe a significant gain in MDL for the corresponding z_n in both cases, indicating that the respective z_n is being protected. Finally, we train ADS using two Bias discriminators and "ADS - $s(h(x))$ (both)" representations achieve the best performance in guarding z_1 & z_2, while performing well on the target task. This shows that ADS framework is scalable and can be leveraged to guard multiple protected attributes simultaneously.

6.2 Efficacy of different losses

We experiment with different configurations of the Scrubber loss $\mathcal{L}_s$ to figure out the efficacy of individual components. We show the experimental results on the Funpedia dataset in Table 6 (with $\lambda_1 = \lambda_2 = 1$). We observe that most leakage in z (increase in prediction F1-score) occur when the entropy loss is removed. Removing δ-loss also results in a slight increase in leakage accompanied by a gain in performance for predicting y. This shows that both losses are important for guarding z.

Empirically, we found that δ-loss is not suitable for binary protected attributes. This is because during training when the Scrubber is encouraged to learn representations that do not have information about z, it learns to encode representations in a manner such that the Bias discriminator predicts the opposite z class. Hence, the information about z is still present and is retrievable using a probing network q. For this reason, we use δ-loss for only Funpedia (λ_2 values in Table 2) where we



(a) Pre-trained (b) After training

Figure 3: UMAP projection of Scrubber output representations $s(h(x))$ from Biographies corpus with profession as "professor". Blue and red labels indicate female and male biographies respectively. (a) Pre-trained BERT representations (b) BERT representations post training in ADS.

considered 3 gender label classes.

6.3 Visualization

We visualize the UMAP (McInnes et al., 2018) projection of Encoder output representations, $h(x)$, in Figure 3. Blue and red labels indicate female and male biographies respectively. Figure 3a and Figure 3b show representations before and after ADS training. In Figure 3a, male and female labeled instances are clearly separated in space. This shows that text representations encode information relating to gender attributes. In Figure 3b, we observe that after training in our adversarial framework both male and female labeled instances are difficult to segregate. This indicates that post training in ADS, it is difficult to identify biography representations on the basis of gender.

7 Conclusion

In this work, we proposed **Ad**versarial **S**crubber (ADS) to remove demographic information from contextual representations. Theoretical analysis showed that under certain conditions, our framework converges without leaking information about protected attributes. We extend previous evaluation metrics to evaluate fairness of representations by using MDL. Experimental evaluations on 8 datasets show that ADS is better at protecting demographic attributes than baselines. We show that our approach is scalable and can be used to remove multiple protected attributes simultaneously. Future work can explore leveraging ADS towards learning fair representations in other NLP tasks.

8 Acknowledgement

This work was supported in part by grants NIH 1R01AA02687901A1 and NSF IIS2133595.

Ethical considerations

We propose ADS, an adversarial framework to prevent text classification modules from taking biased decisions. ADS is intended to be used in scenarios, where the user is already aware of the input attributes they want to protect. ADS can only be trained on data where protected attributes are annotated. It is possible that representations retrieved from ADS, contain sensitive information which were not defined as the protected variables. Even in such a scenario, ADS won't reveal information more than its already available in the dataset. One potential way of misusing ADS would to define relevant features for a task (e.g. experience for a job application) as a protected attribute, then the classification system may be forced to rely on sensitive demographic information for predictions. In such cases, it is possible to flag systems by evaluating the difference in True Positive Rate (TPR) when the protected attribute is changed ($\text{GAP}^{\text{TPR}}_{z,y}$ metric (De-Arteaga et al., 2019)). All experiments were performed on publicly available data, where the identity of author was anonymous. We did not perform any additional data annotation.

References

Kanadpriya Basu, Treena Basu, Ron Buckmire, and Nishu Lal. 2019. Predictive models of student college commitment decisions using machine learning. Data, 4(2):65.

Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. 2020. Language (technology) is power: A critical survey of "bias" in NLP. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 5454–5476, Online. Association for Computational Linguistics.

Su Lin Blodgett, Lisa Green, and Brendan O'Connor. 2016. Demographic dialectal variation in social media: A case study of African-American English. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 1119–1130, Austin, Texas. Association for Computational Linguistics.

Tolga Bolukbasi, Kai-Wei Chang, James Y. Zou, Venkatesh Saligrama, and Adam Tauman Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In

Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain, pages 4349–4357.

John D. Burger, John Henderson, George Kim, and Guido Zarrella. 2011. Discriminating gender on Twitter. In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pages 1301–1309, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Aaron Chalfin, Oren Danieli, Andrew Hillis, Zubin Jelveh, Michael Luca, Jens Ludwig, and Sendhil Mullainathan. 2016. Productivity and selection of human capital with machine learning. American Economic Review, 106(5):124–27.

Maria De-Arteaga, Alexey Romanov, Hanna Wallach, Jennifer Chayes, Christian Borgs, Alexandra Chouldechova, Sahin Geyik, Krishnaram Kenthapadi, and Adam Tauman Kalai. 2019. Bias in bios: A case study of semantic representation bias in a high-stakes setting. In proceedings of the Conference on Fairness, Accountability, and Transparency, pages 120–128.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Emily Dinan, Angela Fan, Ledell Wu, Jason Weston, Douwe Kiela, and Adina Williams. 2020. Multi-dimensional gender bias classification. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 314–331, Online. Association for Computational Linguistics.

Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2018. Measuring and mitigating unintended bias in text classification. In Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society, pages 67–73.

Yanai Elazar and Yoav Goldberg. 2018. Adversarial removal of demographic attributes from text data. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 11–21, Brussels, Belgium. Association for Computational Linguistics.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. 2021. Amnesic probing: Behavioral explanation with amnesic counterfactuals. Transactions of the Association for Computational Linguistics, 9:160–175.

Joel Escudé Font and Marta R Costa-Jussa. 2019. Equalizing gender biases in neural machine translation with word embeddings techniques. arXiv preprint arXiv:1901.03116.

Omar Ghailan, Hoda MO Mokhtar, and Osman Hegazy. 2016. Improving credit scorecard modeling through applying text analysis. institutions, 7(4).

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial networks. arXiv preprint arXiv:1406.2661.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 2733–2743, Hong Kong, China. Association for Computational Linguistics.

Clayton Hutto and Eric Gilbert. 2014. 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.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. OpenReview.net.

Svetlana Kiritchenko and Saif Mohammad. 2018. Examining gender and race bias in two hundred sentiment analysis systems. In Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics, pages 43–53, New Orleans, Louisiana. Association for Computational Linguistics.

Moshe Koppel, Shlomo Argamon, and Anat Rachel Shimoni. 2002. Automatically categorizing written texts by author gender. Literary and linguistic computing, 17(4):401–412.

Yitong Li, Timothy Baldwin, and Trevor Cohn. 2018. Towards robust and privacy-preserving text representations. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 25–30, Melbourne, Australia. Association for Computational Linguistics.

Haochen Liu, Wei Jin, Hamid Karimi, Zitao Liu, and Jiliang Tang. 2021. The authors matter: Understanding and mitigating implicit bias in deep text classification. arXiv e-prints, pages arXiv–2105.

Haochen Liu, Wentao Wang, Yiqi Wang, Hui Liu, Zitao Liu, and Jiliang Tang. 2020. Mitigating gender bias for neural dialogue generation with adversarial learning. In Proceedings of the 2020 Conference on

Empirical Methods in Natural Language Processing (EMNLP), pages 893–903, Online. Association for Computational Linguistics.

Chandler May, Alex Wang, Shikha Bordia, Samuel R. Bowman, and Rachel Rudinger. 2019. On measuring social biases in sentence encoders. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 622–628, Minneapolis, Minnesota. Association for Computational Linguistics.

Leland McInnes, John Healy, and James Melville. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2019. A survey on bias and fairness in machine learning. arXiv preprint arXiv:1908.09635.

Dong Nguyen, Rilana Gravel, Dolf Trieschnigg, and Theo Meder. 2013. " how old do you think i am?" a study of language and age in twitter. In Proceedings of the International AAAI Conference on Web and Social Media, volume 7.

Ji Ho Park, Jamin Shin, and Pascale Fung. 2018. Reducing gender bias in abusive language detection. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 2799–2804, Brussels, Belgium. Association for Computational Linguistics.

Francisco Rangel, 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. Working Notes Papers of the CLEF, 2016:750–784.

Jorma Rissanen. 1984. Universal coding, information, prediction, and estimation. IEEE Transactions on Information theory, 30(4):629–636.

Rachel Rudinger, Chandler May, and Benjamin Van Durme. 2017. Social bias in elicited natural language inferences. In Proceedings of the First ACL Workshop on Ethics in Natural Language Processing, pages 74–79, Valencia, Spain. Association for Computational Linguistics.

Maarten Sap, Dallas Card, Saadia Gabriel, Yejin Choi, and Noah A. Smith. 2019. The risk of racial bias in hate speech detection. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 1668–1678, Florence, Italy. Association for Computational Linguistics.

Danielle Saunders and Bill Byrne. 2020. Reducing gender bias in neural machine translation as a domain adaptation problem. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7724–7736, Online. Association for Computational Linguistics.

Deven Santosh Shah, H. Andrew Schwartz, and Dirk Hovy. 2020. Predictive biases in natural language processing models: A conceptual framework and overview. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 5248–5264, Online. Association for Computational Linguistics.

Emily Sheng, Kai-Wei Chang, Premkumar Natarajan, and Nanyun Peng. 2019. The woman worked as a babysitter: On biases in language generation. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 3407–3412, Hong Kong, China. Association for Computational Linguistics.

Gabriel Stanovsky, Noah A. Smith, and Luke Zettlemoyer. 2019. Evaluating gender bias in machine translation. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 1679–1684, Florence, Italy. Association for Computational Linguistics.

Ben Verhoeven and Walter Daelemans. 2014. CLiPS stylometry investigation (CSI) corpus: A Dutch corpus for the detection of age, gender, personality, sentiment and deception in text. In Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), pages 3081–3085, Reykjavik, Iceland. European Language Resources Association (ELRA).

Ben Verhoeven, Walter Daelemans, and Barbara Plank. 2016. TwiSty: A multilingual Twitter stylometry corpus for gender and personality profiling. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), pages 1632–1637, Portorož, Slovenia. European Language Resources Association (ELRA).

Elena Voita and Ivan Titov. 2020. Information-theoretic probing with minimum description length. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 183–196, Online. Association for Computational Linguistics.

Edson RD Weren, Anderson U Kauer, Lucas Mizusaki, Viviane P Moreira, J Palazzo M de Oliveira, and Leandro K Wives. 2014. Examining multiple features for author profiling. Journal of information and data management, 5(3):266–266.

Richard S. Zemel, Yu Wu, Kevin Swersky, Toniann Pitassi, and Cynthia Dwork. 2013. Learning fair representations. In Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013, volume 28 of JMLR Workshop and Conference Proceedings, pages 325–333. JMLR.org.

Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. 2018. Mitigating unwanted biases with adversarial learning. In Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society, pages 335–340.

A Appendix

A.1 Minimum Description Length

Minimum Description Length (MDL) measures the description length of labels given a set of representations. MDL captures the amount of effort required to achieve a certain probing accuracy, characterizing either complexity of probing model, or amount of data required.

Estimating MDL involves a dataset $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, where x_i's are data representations from a model and y_i's are task labels. Now, a sender Alice wants to transmit labels $\{y_1, \ldots, y_n\}$ to a receiver Bob, when both of them have access to the data representations x_i's. In order to transmit the labels efficiently, Alice needs to encode y_i's in an optimal manner using a probabilistic model $p(y|x)$. The minimum codelength (*Shannon-Huffman code*), required to transmit the labels losslessly is: $\mathcal{L}_p(y_{1:n}|x_{1:n}) = -\sum_{i=1}^{n} \log_2 p(y_i|x_i)$.

There are two ways of evaluating MDL for transmitting the labels $y_{1:n}$ (a) *variational code* - transmit $p(y|x)$ explicitly and then use it to encode the labels (b) *online code* - encodes the model and labels without explicitly transmitting the model. In our experiments, we evaluate the online code for estimating MDL. In the online setting, the labels are transmitted in blocks in n timesteps $\{t_0, \ldots, t_n\}$. Alice encodes the first block of labels $y_{1:t_1}$ using a uniform code. Bob learns a model $p_{\theta_1}(y|x)$ using the data $\{(x_i, y_i)\}_{i=1}^{t_1}$, Alice then transmits the next block of labels $y_{t_1+1:t_2}$ using $p_{\theta_1}(y|x)$. In the next iteration, the receiver trains a new model using a larger chunk of data $\{(x_i, y_i)\}_{i=1}^{t_2}$, which encodes $y_{t_2+1:t_3}$. This continues till the whole set of labels $y_{1:n}$ is transmitted. The total codelength required for transmission using this setting is given as:

$$\mathcal{L}_{online}(y_{1:n}|x_{1:n}) = t_1 \log_2 C - \sum_{i=1}^{n-1} \log_2 p_{\theta_i}(y_{t_i+1:t_{i+1}}|x_{t_i+1:t_{i+1}}) \quad (12)$$

where $y_i \in \{1, 2, \ldots, C\}$. The online codelength $\mathcal{L}_{online}(y_{1:n}|x_{1:n})$ is shorter if the probing model is able to perform well using fewer training instances, therefore capturing the effort needed to achieve a prediction performance.

DATASET	Time/ epoch (min.)
FUNPEDIA	2
WIZARD	1
CONVAI2	14
LIGHT	4
OPENSUB	15
BIOGRAPHIES	260
DIAL	16
PAN16 (gender)	15
PAN16 (age)	15

Table 7: Runtime for each dataset.

A.2 Theoretical Analysis

Proposition. *Minimizing δ-loss is equivalent to increasing the Bias discriminator loss $\mathcal{L}_d$.*
Proof: The δ-loss function can be written as:

$$\delta(o_i) = m_i^T \text{softmax}_{gumble}(o_i) = \frac{\exp\left(\frac{\log o_i^k + g_k}{\tau}\right)}{\sum_j \exp\left(\frac{\log o_i^j + g_j}{\tau}\right)} \quad (13)$$

where o_i^j is the raw logit assigned to the j^{th} output class, the true output class is $k = z_i$ and g_j, g_k are i.i.d samples from Gumble(0,1) distribution. The cross entropy loss of the bias discriminator $\mathcal{L}_d$ can be written as:

$$\mathcal{L}_d = -\log \frac{\exp(o_i^k)}{\sum_j \exp(o_i^j)} \quad (14)$$

The gumble softmax generates a peaked version of the normal softmax distribution. But the individual gumble softmax logit values (Equation 13) are still proportional to vanilla softmax logits (Equation 14): $\delta(o_i) \propto \frac{\exp o_i^k}{\sum_j \exp o_i^j}$. Then, bias discriminator loss $\mathcal{L}_d$ can be written as:

$$\mathcal{L}_d \propto -\log \delta(o_i) \quad (15)$$

Therefore, minimizing $\delta(o_i)$ increases $\mathcal{L}_d$.

A.3 Implementation Details

All experiments are conducted in PyTorch framework using Nvidia GeForce RTX2080 GPU with

Dataset	Pre-trained $h(x)$		w/o adversary $h(x)$		ADS $h(x)$		ADS $s(h(x))$	
	$\overrightarrow{\text{MDL}}_z$	$\overrightarrow{\text{MDL}}_y$	$\overrightarrow{\text{MDL}}_z$	$\overrightarrow{\text{MDL}}_y$	$\overrightarrow{\text{MDL}}_z$	$\overrightarrow{\text{MDL}}_y$	$\overrightarrow{\text{MDL}}_z$	$\overrightarrow{\text{MDL}}_y$
FUNPEDIA	1.03	1.94	1.29	0.12	1.48	0.43	1.73	0.45
WIZARD	1.08	2.15	1.47	0.06	1.84	0.09	1.95	0.06
CONVAI2	1.46	1.94	1.58	0.09	1.94	0.16	1.93	0.16
LIGHT	1.21	2.28	1.44	0.21	1.89	0.43	1.92	0.42
OPENSUB	0.92	2.03	1.49	0.12	1.77	0.18	1.78	0.18
BIOGRAPHIES	0.11	1.94	1.74	0.01	1.73	0.01	1.74	0.01
DIAL	1.46	1.81	1.06	0.60	1.65	0.31	1.75	0.34
PAN16 (gender)	1.87	1.62	1.67	0.03	1.90	0.04	1.96	0.05
PAN16 (age)	1.89	1.64	1.85	0.03	1.89	0.03	1.97	0.04

Table 8: Probing performance of representations retrieved from different settings in terms of $\overrightarrow{\text{MDL}}$.

12GB memory. We use an off-the-shelf MLPClassifer from *sklearn*[8] as our *probing network* q. ADS has a total of 110M parameters (all 4 modules combined). The average runtime per epoch for each dataset is reported in Table 7.

A.4 Measuring Fairness in Representations

MDL scales linearly with the dataset size (Equation 12), therefore making it hard to compare across different datasets. In order to make it comparable, we measure a normalized description length measure for transmitting 1000 labels:

$$\overrightarrow{\text{MDL}} = \frac{1000 \times \text{MDL}}{|\mathcal{D}|} \tag{16}$$

$|\mathcal{D}|$ is the dataset size. Performance using this measure are reported in Table 8 for all datasets. In all experiments we report the MDL required for transmitting the labels in the training set.

[8] https://scikit-learn.org/