Dynamically Disentangling Social Bias from Task-Oriented Representations with Adversarial Attack

Liwen Wang[1*], **Yuanmeng Yan**[1*], **Keqing He**[1,2], **Yanan Wu**[1], **Weiran Xu**[1]
[1]Beijing University of Posts and Telecommunications, Beijing, China
[2]Meituan Group, Beijing, China
{w_liwen,yanyuanmeng,yanan.wu,xuweiran}@bupt.edu.cn
{hekeqing}@meituan.com

Abstract

Representation learning is widely used in NLP for a vast range of tasks. However, representations derived from text corpora often reflect social biases. This phenomenon is pervasive and consistent across different neural models, causing serious concern. Previous methods mostly rely on a pre-specified, user-provided direction or suffer from unstable training. In this paper, we propose an adversarial disentangled debiasing model to dynamically decouple social bias attributes from the intermediate representations trained on the main task. We aim to denoise bias information while training on the downstream task, rather than completely remove social bias and pursue static unbiased representations. Experiments show the effectiveness of our method, both on the effect of debiasing and the main task performance.

1 Introduction

Supervised neural networks have achieved remarkable success in a wide range of natural language processing (NLP) tasks. The fundamental capability of these neural models is to learn effective feature representations (Bengio et al., 2013) for the downstream prediction task. Unfortunately, the learned representations frequently contain undesirable biases with respect to things that we would rather not use for decision making. We refer to such inappropriate factors as protected attributes (Elazar and Goldberg, 2018a). Biased information has serious real-world consequences. For example, concerns have been raised about automatic resume filtering systems giving preference to male applicants when the only distinguishing factor is the applicants' gender (Sun et al., 2019). In this paper, we focus on social bias, such as gender bias which is the preference or prejudice towards one gender over the other (Moss-Racusin et al., 2012), race bias and age bias.

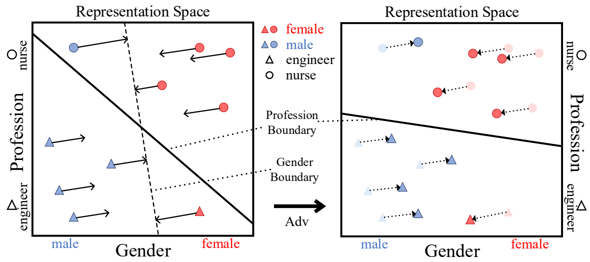


Figure 1: A demonstration of dynamically disentangling gender bias from the representations of downstream representations. The left figure shows a biased profession classifier where the model prefers men as engineers and women as nurses. Our adversarial disentanglement model makes the example features approach the boundary of the gender classifier. Therefore, the boundary of the profession classifier in the right figure shifts closer to the horizontal state where gender information is decoupled from the representations in the main task.

From the perspective of the debiasing target, previous debiasing works can be approximately classified into two types, word embedding (Bolukbasi et al., 2016; Caliskan et al., 2017; Zhao et al., 2018; Manzini et al., 2019; Wang et al., 2020; Kumar et al., 2020) and sentence embedding (Xu et al., 2017; Elazar and Goldberg, 2018a; Zhang et al., 2018; Ravfogel et al., 2020). The former aims to reduce the gender bias in word embedding, either as a post-processing step (Bolukbasi et al., 2016) or as part of the training procedure (Zhao et al., 2018). The latter focuses on removing these protected attributes from the downstream intermediate representations (Elazar and Goldberg, 2018a; Ravfogel et al., 2020). In this paper, we consider the latter setting and focus on how to mitigate undesirable social bias from the encoded representations without hurting the performance of the main task.

In terms of debiasing methods, previous models are either based on projection on a pre-specified, user-provided direction (Bolukbasi et al., 2016) or null-space (Xu et al., 2017; Ravfogel et al., 2020),

*The first two authors contribute equally. Weiran Xu is the corresponding author.

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 3740–3750
June 6–11, 2021.

or on adding an additional gender discriminator (Xie et al., 2017; Elazar and Goldberg, 2018a). The former first trains an intermediate feature extractor on the main task, then using a separate projection method to remove social bias from the representations, finally fine-tuning on the main task. The debiasing procedure can be regarded as static because of no direct interaction between the main task and the debiasing task. Therefore, these methods have no guarantee that the representations for predicting the main task do not contain any bias information. Existing work, (Gonen and Goldberg, 2019), has shown that these methods only *cover up* the bias and that in fact, the information is deeply ingrained in the representations. Compared to these static debiasing methods, gender discriminator based methods (Elazar and Goldberg, 2018a; Zhang et al., 2018) use the traditional generative adversarial network (GAN) (Goodfellow et al., 2014) to distinguish protected gender attributes from encoded representations. However, they are notoriously hard to train (Ganin and Lempitsky, 2015). Elazar and Goldberg (2018a) has shown that the complete removal of the protected information is nontrivial: even when the attribute seems protected, different classifiers of the same architecture can often still succeed in extracting it. Hence, we aim to dynamically disentangle the social bias from the encoded representations while jointly training on the main task in a more stable way, rather than directly remove protected attributes. In fact, we show that bias information always remains even after adversarial debiasing and can be reconstructed from the encoded representations. The main goal of debiasing is to prevent downstream models from utilizing these social bias in the representations, that is, dynamic disentanglement instead of complete removal, as Fig 1 displays.

In this paper, we propose an adversarial disentangled debiasing model to dynamically decouple social bias attributes from the intermediate representations trained on the main task. Our motivation is to denoise bias information while training on the downstream task, rather than completely remove social bias and pursue static unbiased representations. Previous works (Elazar and Goldberg, 2018a; Gonen and Goldberg, 2019) show that even debiasing models achieve high fairness (Hardt et al., 2016), a fair amount of protected information still remains and can be extracted from the encoded representations. We argue that one can hardly re-

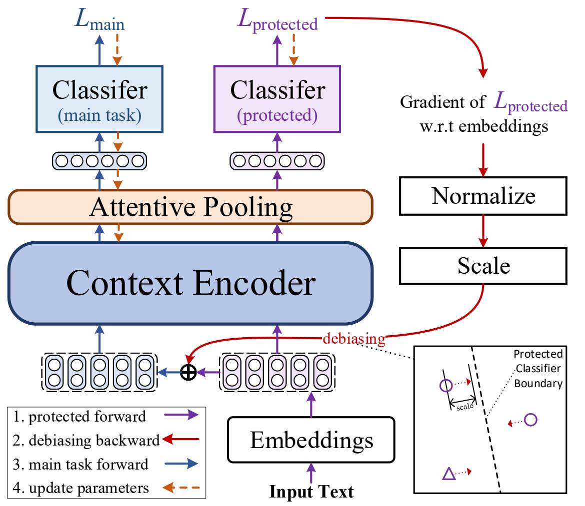


Figure 2: The overall architecture of our proposed approach.

move all gender or race directions in the latent space but only preserve bias-free prediction on the downstream task. Specifically, we use a protected attribute classifier to generate model-agnostic adversarial worst-case perturbations to the representations in the direction that significantly increases the classifier's loss. Then we apply the perturbations to train the model of the downstream task end-to-end. The main difference between our method and GAN-based counterparts is that GANs suffer from unstable training for the two-stage min-max procedure but our method directly computes gradient-based perturbations to disentangle bias information from the representations. We hope to provide new insights and directions towards solving social bias issues. [1]

2 Approach

2.1 Problem Formulation

Our main goal is to disentangle protected attributes from the representations of downstream tasks so that biased information can not affect the decision of the model on the main task. In other words, we aim to achieve fairness by equalizing the opportunity (Hardt et al., 2016) between individuals with different protected attributes (e.g. gender/race). Given a set of input samples x_i, and corresponding discrete attributes $Z, z_i \in \{1, \ldots, k\}$ (e.g. gender or race) [2], we aim to learn unbiased representations

[1]Our source code is available at https://github.com/W-lw/debias_adv.

[2]Although we focus on the discrete protected attributes in this paper, our method can be also applied to continuous attributes.

$\boldsymbol{h}_i \in \mathbb{R}^d$, so that z_i can pose as minimal negative effect as possible on the main task performance.

2.2 Overall Architecture

Fig 2 shows the overall architecture of our proposed method, including four core steps: protected forward, debiasing backward, main task forward, and update parameters. (1) **protected forward**: We first pre-train a protected attribute classifier then compute the classification cross-entropy loss $\mathcal{L}_{protected}$ for each input sample x. (2) **debiasing backward**: We maximize the loss $\mathcal{L}_{protected}$ of the protected attribute classifier to obtain the adversarial decoupling perturbation $\boldsymbol{\delta}$. (3) **main task forward**: Then we sum the original input x and perturbation $\boldsymbol{\delta}$ to get a new adversarial sample x_{adv}. We forward the sample x_{adv} to the main task classifier to compute the loss $\mathcal{L}_{main}$ of the downstream task. (4) **update parameters**: Finally, the overall model is updated by the sum of two losses $\mathcal{L}_{protected}, \mathcal{L}_{main}$. We will dive into the details of each procedure in the following section.

2.3 Adversarial Semantic Disentanglement

Protected Forward In Fig 2, we adopt BiLSTM as the shared context encoder by the main task classifier and protected attribute classifier. We first feed each token to an embedding layer to get token embedding $\boldsymbol{e}$, then a BiLSTM encoder is adopted to get the context-aware representation h_i for each token x_i. Then, we use an attentive pooling layer to calculate the sentence embedding h. After that, a fully-connected layer followed by a softmax output layer is used to predict the protected attribute $\hat{y}_i$. Finally, we can get the classification cross-entropy loss $\mathcal{L}_{protected}$. [3] In the experiment, we observe that pre-training the protected attribute classifier can effectively accelerate the whole training progress of debiasing. We also demonstrate that jointly training the protected attribute classifier and the main task classifier achieves superior performance in Section 4.2.

Debiasing Backward This is the primary step of our adversarial semantic disentanglement. Our main idea is to perform adversarial attacks (Goodfellow et al., 2015; Kurakin et al., 2016; Miyato et al., 2016; Jia and Liang, 2017; Zhang et al., 2019; Ren et al., 2019) to dynamically decouple social bias attributes from the intermediate representations trained on the main task. Specifically, we need to compute a worst-case perturbation $\boldsymbol{\delta}$ that maximizes the original classification cross-entropy loss $\mathcal{L}_{protected}$ of the protected attribute classifier:

$$\boldsymbol{\delta} = \underset{\|\boldsymbol{\delta}'\| \le \epsilon}{\arg\max} \quad \mathcal{L}_{protected}\left(\boldsymbol{\theta}, x + \boldsymbol{\delta}'\right) \tag{1}$$

where $\boldsymbol{\theta}$ represents the parameters of the protected attribute classifier and $\boldsymbol{x}$ denotes a given sample. ϵ is the norm bound of the perturbation $\boldsymbol{\delta}$. However, due to model complexity, accurate computation for $\boldsymbol{\delta}$ is costly and inefficient. Similar to Vedula et al. (2020) and Ru et al. (2020), we apply Fast Gradient Value (FGV) (Rozsa et al., 2016) to approximate a worst-case perturbation $\boldsymbol{\delta}$:

$$\boldsymbol{\delta} = \epsilon \frac{g}{||g||}; \text{where } g = \nabla_{\boldsymbol{e}} \mathcal{L}(f(\boldsymbol{e}; \boldsymbol{\theta}), Y) \tag{2}$$

where f represents the protected attribute classifier. We perform normalization to g and then use a small ϵ to ensure the approximate is reasonable. Section 4.3 validates a proper value of ϵ can balance the debiasing effect and the main task performance. Finally, we can obtain the pseudo adversarial sample $x_{adv} = x + \boldsymbol{\delta}$. Intuitively we aim to obtain a debiased representation x_{adv} by confusing the protected attribute classifier. Thus, the main task classifier can make a fair decision conditioned on the disentangled representation.

Main Task Forward After obtaining the pseudo adversarial sample x_{adv}, we forward the sample x_{adv} to the main task classifier to compute the loss $\mathcal{L}_{main}$ of the downstream task, similar to protected forward. We find the location of adding adversarial perturbation plays a role in debiasing performance in Section 4.4. In a nutshell, adding noise to the word embedding layer achieves the best debiasing performance.

Update Parameters Finally, we apply the two classification objectives to update the parameters of the model as the dashed lines in Fig 2 show. Note that the loss $\mathcal{L}_{protected}$ of the protected attribute classifier only updates the MLP and softmax layers while the loss $\mathcal{L}_{main}$ of the main task classifier updates all the model parameters, including the low-level encoding layers. The setting aims to avoid the negative effect of the protected attribute classifier on main task performance.

3 Experiments

3.1 Setup

Datasets Following the setup of (Ravfogel et al., 2020), we test the performance of our debiasing

[3] If the protected attribute is continuous, we can apply the regression objectives.

	Sentiment (Main Task)				TPR-GAP (Debias)			
Ratio	Original	INLP	Random Noise	Ours	Original	INLP	Random Noise	Ours
0.5	0.76	0.75	0.75	0.75	0.14	0.12	0.14	**0.09**
0.6	0.75	0.71	0.73	0.72	0.23	0.18	0.17	**0.11**
0.7	0.74	0.65	0.72	0.72	0.31	0.16	0.26	**0.10**
0.8	0.71	0.62	0.71	0.73	0.40	0.16	0.37	**0.09**

Table 1: The Sentiment scores (in accuracy, higher is better) and TPR differences (lower is better) as a function of the ratio of tweets written by black individuals in the positive-sentiment class.

method on the dialectal tweets (DIAL) corpus collected by Blodgett et al. (2016) in a controlled setup, and the biography corpus (De-Arteaga et al., 2019) in a wild setup. The dialectal tweets corpus consists of 59.2 million tweets, where each tweet contains "race" information, and emojis correspond with specific emotion groups. According to the label of race and sentiment, we split the data into four classes: African American English (AAE) speaker with "happy" sentiment, Standard American English (SAE) speaker with "happy" sentiment, AAE speaker with "sad" sentiment and SAE speaker with "sad" sentiment. Following (Elazar and Goldberg, 2018b), we filter the corpus and 176K tweets left (44k for each class). Then we divide them into 40k samples for training, 2k for developing, and 2k for testing, following (Ravfogel et al., 2020). In the controlled setup, we introduce a bias ratio relevant to the sentiment and race to control the imbalance proportion of samples in four groups, following (Ravfogel et al., 2020). e.g., in the 0.8 condition, the AAE class contains 80% happy / 20% sad samples, while the SAE class contains 80% sad / 20% happy samples. And in the 0.5 conditions, all four categories contain the same number of samples. In all experiments, the unbalance factor of the development set and test set is set to 0.5.

The biography corpus contains 393,423 biographies, the corresponding professions (28 classes) labels and gender (protected attributes) labels. We split the dataset into 255,710, 39,369, 98,344 samples for training, validation and testing, as consistent with (De-Arteaga et al., 2019; Ravfogel et al., 2020).

Baselines We compare our model with these baselines as follow:

- **Original** is the main task classifier without any debiasing procedure as a baseline.

- **INLP** (Ravfogel et al., 2020) is a linear debiasing method, which removes the protected information from neural representations by iterative training the linear classifiers which predict the protected attributes. [4]

- **Random Noise** replaces the debiasing perturbation generated by the protected classifier with random noise.

Implementation Details To demonstrate the effectiveness of our method, we use the same model structure of the main task (sentiment classification) as (Ravfogel et al., 2020), where the DeepMoji encoder (Felbo et al., 2017) and an one-hidden-layer MLP constitute the classifier. Besides, for simplicity, we use the same structure of classifier for predicting protected attributes. Both the unbalanced training data and the pre-trained DeepMoji model which has been proven that encodes demographic information would lead the downstream MLP classifier to make biased predictions. We then perform debiasing training for the main-task model following the process described in section 2.3 on the imbalanced training set with the imbalance factor and test the debiased model on the balanced test set.

Besides, we follow (Ravfogel et al., 2020) to evaluate our debiasing method on the biography corpus as a wild setup to verify the validity of our method in a less artificial setting. In this wild set up, we construct a similar model structure to the DeepMoji encoder, with a two-layer bidirectional RNN as the encoder, except for the attention operation. There are two input representation types of the encoder: FastText and BERT (Devlin et al., 2019). In the FastText experiments, we directly use the trained word embedding that provided by (Ravfogel et al., 2020), to represent each biography as a sequence of vectors. And in the BERT experiments, we use BERT as a sequence-to-sequence encoder

[4]Note that the original results reported in the published version contain some mistakes. We rerun the updated evaluation scripts according to the official code and report all the results for a fair comparison.

to obtain the representation of each word in the sentence. Then we feed the sentence representations into the model and perform the debiasing training process.

For all the experiments, we train and test our model on single 2080Ti GPU, and we use AllenNLP framework (Gardner et al., 2017) to implement our model. The hidden size of the 1-hidden-layer MLP classifier used in all of the above experiments is set to 300. In a controlled experiment, our debiasing method takes an average of ten minutes to run, and the total parameters of our models are 23M, including a DeepMoji encoder, a main task classifier, and a protected classifier. In the wild experiment, the model size of the FastText experiment is 127M, which takes an average of 15 minutes to run. While the model size of the BERT experiment is 114M, and it takes an average of 55 minutes to run, due to the use of BERT to encode the sentences. It's worth mentioning that our method converges with only one or two epochs, which is faster than other debiasing methods. In practice, we empirically find that the debiasing performance can reach the best when the L2-Norm of perturbation is between 1/3 and 2/3 of the corresponding disturbed vectors' L2-Norm. For example, in the first experiment, the L2-Norm size of the embedding vector is around 4, then we could set the normalized scale to (1.2, 1.8).

Metrics To evaluate the bias in the model, following (Ravfogel et al., 2020; De-Arteaga et al., 2019), we calculate TPR-GAP to measure the difference (GAP) in the True Positive Rate (TPR) between the groups with different protected attributes which can reflect the unfairness existing in NLP models:

$$TPR_{p,y} = P[\hat{Y} = y | P = p, Y = y] \quad (3)$$

$$GAP^{TPR}_{p,y} = TPR_{p,y} - TPR_{p',y} \quad (4)$$

where y is the main task label of the input representation X, and p, p' denote the protected attribute P's two values. Then we use TPR-GAP to measure the degree of bias, which calculate the root-mean square of $GAP^{TPR}_{p,y}$ over all main task label y:

$$GAP^{TPR,RMS} = \sqrt{\frac{1}{|N|} \sum_{y \in N} (GAP^{TPR}_{p,y})^2} \quad (5)$$

where N is the label set of all main task (sentiment or profession). De-Arteaga et al. (2019) did the experiment on the biography corpus, and proved the indicator $GAP^{TPR}_{p,y}$ have a strong correlation with the percentage of a certain gender group in different profession y, therefore $GAP^{TPR,RMS}$ can reflect an overview of bias across all different main attributes. We use $GAP^{TPR,RMS}$ to measure the bias existing in the models.

		FastText	BERT
Accuracy (profession)	Original	78.1	80.9
	INLP	73.0	75.2
	Ours	80.1	77.8
TPR-GAP	Original	0.184	0.184
	INLP	0.089	0.095
	Ours	**0.082**	**0.092**

Table 2: Fair classification on the Biographies corpus.

3.2 Main Results

Table 1 displays the experimental results on the DIAL dataset under different ratios of data imbalance proportion which can reflect the degree of dataset bias. We analyze the results from two perspectives, TPR-GAP (Debias) and Sentiment (Main Task). For TPR-GAP (Debias), our method consistently outperforms other baselines under all ratios, especially on the more biased dataset. It demonstrates the effectiveness of our proposed adversarial semantic disentanglement. We also observe Random Noise can hardly mitigate social bias which confirms the necessity of the protected attribute classifier. For the performance of the main sentiment classification task, our method reaches close to the original baseline while INLP suffers from a severe drop under a large ratio. The results prove that our method can better avoid the negative effect of the debiasing procedure on main task performance. To further evaluate the debiasing effect, we also show the results of the wild biography classification dataset in Table 2. Results show that our method both achieves superior performance than other baselines on Accuracy of the main task and TPR-GAP of debiasing. Compared to the significant improvements on the DIAL dataset, we hypothesize that the bias degree of the dataset makes a difference to the range of improvements.

4 Qualitative Analysis

4.1 Fixed Encoder vs. Non-fixed Encoder

In previous works, it is common to pre-train the sentence encoder in advance and keep the encoder fixed while applying the debias algorithm. However, it is unclear whether this conventional experiment setup is applicable to our approach. Since

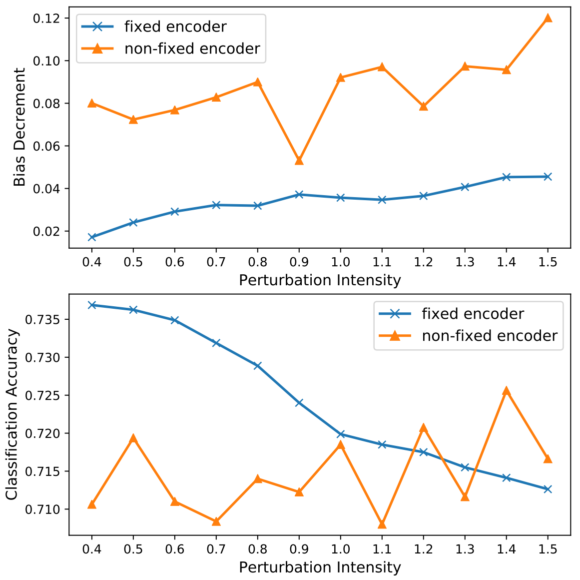


Figure 3: Performance comparison between fixed and non-fixed encoders. The above figure shows the bias decrement under different perturbation intensities while the below figure shows the classification performance of the main task. The experiments are adopted in the DIAL dataset, with the bias ratio set to 0.6.

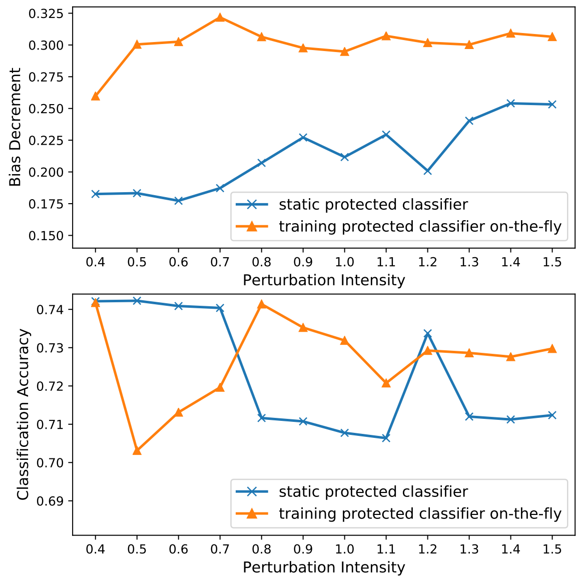


Figure 4: Performance comparison between the static protected classifier and the training on-the-fly protected classifier. The above figure shows the bias decrement under different perturbation intensities while the below figure shows the classification performance of the main task. The experiments are adopted in the DIAL dataset, with the bias ratio set to 0.8.

our approach dynamically generates perturbation to decouple social bias from context via adversarial attacks, we expect the non-fixed encoder to generate perturbation of higher quality. To check this, we conduct two groups of experiments in the DIAL dataset, where one group uses a fixed encoder while the other group keeps the contextual encoder trainable. Note that we set the bias ratio to 0.6 in both two groups of experiments.

Fig 3 shows the experimental results. In Fig 3 above, we observe that our approach with the non-fixed encoder consistently achieves better debias effectiveness compared to the fixed encoder counterpart with a large margin. When the perturbation intensity increases, both experimental settings achieve an increasingly better debias effect.

On the other hand, as shown in Fig 3 below, the fixed encoder approach suffers a severe performance drop in classification accuracy with increasing perturbation intensity. Meanwhile, the classification accuracy under the non-fixed encoder setting is still increasing, and even outperforms the fixed encoder one when a relatively large perturbation intensity is applied. We argue that, with a non-fixed encoder, our approach can learn a high-quality perturbation for representation debias, and meanwhile continuously optimize for the main task.

4.2 Protected Classifier: "Static" vs. Training on-the-fly

As discussed in the previous section, our proposed adversarial disentangled debiasing method requires the protected classifier to learn an accurate decision boundary of the protected attributes, such that the debiasing perturbation approximates the direction that mostly eliminates the model's discrimination of the protected attributes. Naturally, we have two options: either fix the parameters of protected classifier to generate the relatively static debiasing perturbation, or train the protected classifier on-the-fly during the main classifier training process to offer a relatively dynamic perturbation.

To verify which one performs better, we adopt two groups of experiments. In the "static" setting, we keep the parameters of the protected classifier fixed. Whether the parameters of the encoder are fixed or not, the debiasing perturbation generated by the protected classifier would be relatively static. It's worth noting that if the parameters of the encoder are fixed, the debiasing perturbation would be totally static. While in the training on-the-fly setting, we reserve the gradient of the protected classifier and update its parameters together with the main task model (context encoder and main task classifier). According to the conclusions in

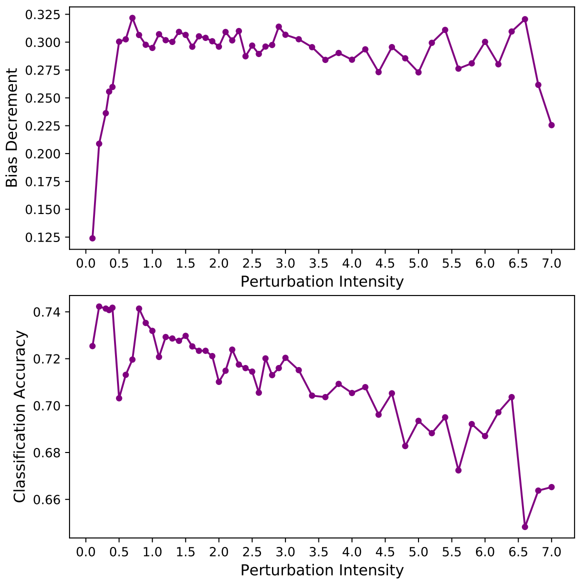


Figure 5: The debias effectiveness (above) as well as the classification accuracy on the main task (below) of our proposed approach in the DIAL dataset, with the perturbation intensity increases from 0.1 to 7.0. We set the bias ratio to 0.8 and all parameters trainable.

section 4.1, we make the context encoder trainable in both settings and use the same objective to train the main classifier.

The results are displayed in Fig 4. We can find that both settings have the ability to debiasing in the DIAL dataset, showing the effectiveness of our approach in both settings. However, the training on-the-fly strategy consistently outperforms the "static" strategy under various perturbation intensities. We hypothesize that the difference is mainly because under the training on-the-fly strategy, the protected classifier will have a chance to adjust the decision boundary when the context encoder updates, and thus continuously generates better dynamic debiasing perturbation via adversarial attacks.

4.3 Influence of Perturbation Intensity

To explore how the perturbation intensity influences the debias effectiveness and the performance of main task, we run multiple experiments with only changing the perturbation intensity. We experiment with a wide range of perturbation intensity, from 0.1 to 7.0.

The experimental results are illustrated in Fig 5. From the figure above, we find that the bias decrement rapidly increases at the beginning period with the intensity increasing from 0.1 to 0.7. Then, between a wide range from 0.7 to 6.6, the bias decrement keeps relatively stable, oscillate in a small range of 0.275 - 0.325, reflecting the stability of our approach. However, when the perturbation intensity exceed some threshold (6.6 in this case), the bias decrement drops again. Meanwhile, with the perturbation intensity increasing, the classification accuracy of main task keeps falling (figure below), indicating that the perturbation with high intensity will also disturb the main task, leading to a low classification accuracy. The result provides a principle of how to choose a suitable perturbation intensity - the minimal intensity while effective enough for debiasing.

DIAL bias ratio →		0.5	0.6	0.7	0.8
Accuracy	Original	0.75	0.75	0.74	0.71
	To sent emb	0.75	0.72	0.72	0.72
	To word emb	0.73	0.72	0.72	0.73
TPR-GAP	Original	0.14	0.23	0.31	0.40
	To sent emb	**0.09**	0.14	0.19	0.21
	To word emb	**0.09**	**0.11**	**0.10**	**0.09**

Table 3: Analysis on which representation space is best for debiasing. "To sent emb" indicates the perturbation is added to the sentence embedding space, while "To word emb" indicates the perturbation is added to the word embedding space. The perturbation intensity is set to 0.7.

4.4 Which Representation Space to Apply Debiasing

Another pivotal consideration for our dynamically disentangling approach is - which representation space should we add the perturbation to? Typically, we have two choices: a) adding the perturbation to the sentence embedding space or b) adding the perturbation to the word embedding space. The sentence embedding is closer to the output space with the key information condensed into a single vector, while the word embedding is closer to the input side, keeping separated for each token. To check out which one performs better for social debiasing, we conduct experiments in the DIAL dataset with different bias ratio.

Table 3 illustrates the experiment results. Compared the result of "To sent emb" to "To word emb", we found adding the perturbation to word embedding space often gains better debiasing results, especially when the bias ratio of the dataset is large. For example, when the bias ratio is 0.8, adding to word embedding space achieves a $GAP^{TPR,RMS}$ of 0.09, while adding to sentence embedding space achieves 0.21. We believe that, when applying our debiasing approach to a deeper representation

DIAL bias ratio →		0.5	0.6	0.7	0.8
Accuracy	Original	0.75	0.75	0.74	0.71
	Entropy	0.74	0.71	0.70	0.72
	Cross entropy	0.75	0.72	0.72	0.73
TPR-GAP	Original	0.14	0.23	0.31	0.40
	Entropy	0.13	0.15	0.17	0.17
	Cross entropy	**0.09**	**0.11**	**0.10**	**0.09**

Table 4: Experimental results on accuracy and debiasing effect with different objectives of the protected classifier. We respectively apply the Cross-Entropy loss and the Entropy loss to the protected classifier when calculating the objective of the protected classifier for generating the perturbation for debiasing. Note that the protected classifier is pre-trained and fixed, and the entropy loss doesn't require ground truth protected attributes during the training of the main task.

space, the perturbation is also context-aware (since the context encoder is also related when calculating the gradient) and thus more dynamic for the complex data distribution.

4.5 Cross-Entropy vs. Entropy

As mentioned in Section 2.3, we need to calculate a cross-entropy loss $\mathcal{L}_{protected}$ to generate the debiasing perturbation via FGV. Thus, during the training of the main task, we must obtain the protected attribute for each training example to calculate the cross-entropy loss. This severely limits the usefulness of our approach, as it may be difficult to obtain the ground truth protected attribute when training the main task. To this end, we also propose to use the entropy loss (Zheng et al., 2020) to substitute the cross-entropy loss:

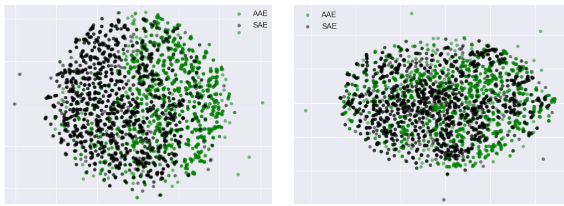
$$\mathcal{L}_{protected} = -\mathcal{H}(P(\boldsymbol{y}_{protected}|x)) \tag{6}$$

where $\mathcal{H}$ indicates the Shannon entropy and $P(\mathbf{y}_{protected}|x)$ is the distribution output by protected classifier. This objective forces the protected classifier to obtain high entropy, which means the classifier is not confident and almost distributed uniformly across all values of the protected attributes.

In Table 4, we compare the debiasing effectiveness of using entropy with cross-entropy. From the table, we observe that using the entropy objective also works for debiasing as the TPR-GAP also drops compared with the baseline. However, the debiasing effect still can't exceed our approach with cross-entropy. This seems reasonable since the cross-entropy objective introduces extra information about the protected attribute. With the extra supervision signal, our approach generates perturbation towards a more precise direction for eliminating the representation of the protected attributes.

DIAL bias ratio →		0.5	0.6	0.7	0.8
RIM	INLP	0.143	0.164	0.362	0.473
	Ours	0.357	0.482	0.650	0.814

Table 5: The debiasing effect under our proposed Relative Improve Metric (RIM). We show that our approach is far beyond the INLP under the evaluation of RIM.



(a) Original (b) Ours

Figure 6: t-SNE projection of BiLSTM sentence representations for the positive sentiment. The left represents the baseline and the right represents our method. We display the representation distribution of different races in the latent space. Different races are colored in the figure.

4.6 Performance on Different Bias Ratio

To more clearly show the performance differences of our model over data sets with varying degrees of bias, we introduce a new metric named Relative Improve Metric (RIM):

$$RIM = \frac{Acc' - Acc}{Acc} + \frac{GAP - GAP'}{GAP} \tag{7}$$

where Acc and Acc' represent the main task accuracy of the model before and after debiasing respectively, and GAP, GAP' represent the TPR-GAP indicator of the model before and after debiasing respectively. RIM could synthetically reflect the stability of the main task and the debiasing performance of a debiasing method. We calculate the RIM indicator of our model and INLP based on the results in Table 1, and the new results are shown in Table 5. We can observe that the stronger bias in the dataset, the better performance of the two methods. Besides, we can find that our debiasing method is more robust.

4.7 Visualization

To better understand the effectiveness of our method, we display a feature visualization of sentence representations in Fig 6. We can observe that the different race classes are no longer linearly separable after debiasing. Therefore, downstream tasks can not make decisions conditioned on the race information in the representations.

5 Related Work

The objective of controlled removal of specific types of information from neural representations is tightly related to the task of disentanglement of the representations (Bengio et al., 2013), that is, controlling and separating the different kinds of information encoded in them. Previous models are either based on projection on a pre-specified, user-provided direction (Bolukbasi et al., 2016) or null-space (Xu et al., 2017; Ravfogel et al., 2020), or adding an additional gender discriminator (Xie et al., 2017; Elazar and Goldberg, 2018a), or the impact of data decisions (Beutel et al., 2017). The former first train an intermediate feature extractor on the main task, then use a separate projection method to remove social bias from the representations, finally finetune on the main task. Compared to these static debiasing methods, gender discriminator based methods (Elazar and Goldberg, 2018a; Zhang et al., 2018) use the traditional generative adversarial network (GAN) (Goodfellow et al., 2014) to remove protected gender attributes from encoded representations. However, they are notoriously hard to train (Ganin and Lempitsky, 2015). Elazar and Goldberg (2018a) has shown that the complete removal of the protected information is nontrivial: even when the attribute seems protected, different classifiers of the same architecture can often still succeed in extracting it. Therefore, in this paper, we aim to dynamically disentangle the social bias from the encoded representations while jointly training on the main task in a more stable way, rather than directly remove protected attributes. The main goal of debiasing is to prevent downstream models from utilizing these social biases in the representations, that is, dynamic disentanglement instead of complete removal.

6 Conclusion

In this paper, we focus on removing social bias in representation learning. We argue that the main goal of debiasing is to prevent downstream models from utilizing these social biases in the representation, that is, dynamic disentanglement instead of complete removal. Therefore, we propose an adversarial disentangled debiasing model to dynamically decouple social bias attributes from the intermediate representation trained on the main task. We perform extensive experiments and analysis to demonstrate the effectiveness of our method. We hope to provide new insights and directions towards solving social bias.

7 Broader Impact

In recent years, neural network based models have demonstrated remarkable performance in many natural language processing tasks and thus have been applied to a wide range of real-world applications. However, a lot of works reveal that such models are easily affected by social bias and thus makes incorrect and biased decisions. In domains with the greatest potential for societal impacts, using such biased models for real-world applications is dangerous and faces many problems such as human morality. The social bias implicit in the natural language processing model may be exposed and become a social hot spot, thus becoming an unstable factor that causes social unrest. Meanwhile, some existing debiasing methods, although able to slightly reduce bias in such model, often cause great damage to model performance in the main task, thus difficult to be applied in practice. This work proposes a new adversarial training method for end-to-end debiasing. Due to the robustness of the adversarial attack, the model can eliminates bias without losing much performance.

8 Acknowledgements

This work was partially supported by National Key RD Program of China No. 2019YFF0303300 and Subject II No. 2019YFF0303302, DOCOMO Beijing Communications Laboratories Co., Ltd, MoE-CMCC "Artifical Intelligence" Project No. MCM20190701.

References

Yoshua Bengio, Aaron C. Courville, and P. Vincent. 2013. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35:1798–1828.

Alex Beutel, Jilin Chen, Zhe Zhao, and Ed H. Chi. 2017. Data decisions and theoretical implications when adversarially learning fair representations.

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 A. Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *ArXiv*, abs/1607.06520.

A. Caliskan, J. Bryson, and A. Narayanan. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356:183 – 186.

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. *Proceedings of the Conference on Fairness, Accountability, and Transparency - FAT* '19*.

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.

Yanai Elazar and Y. Goldberg. 2018a. Adversarial removal of demographic attributes from text data. *ArXiv*, abs/1808.06640.

Yanai Elazar and Yoav Goldberg. 2018b. 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.

Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. 2017. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1615–1625, Copenhagen, Denmark. Association for Computational Linguistics.

Yaroslav Ganin and V. Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In *ICML*.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. 2017. Allennlp: A deep semantic natural language processing platform.

H. Gonen and Y. Goldberg. 2019. Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them. In *NAACL-HLT*.

Ian J. Goodfellow, Jean Pouget-Abadie, M. Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *NIPS*.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572.

M. Hardt, E. Price, and Nathan Srebro. 2016. Equality of opportunity in supervised learning. In *NIPS*.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. *arXiv preprint arXiv:1707.07328*.

V. Kumar, Tenzin Singhay Bhotia, and Tanmoy Chakraborty. 2020. Nurse is closer to woman than surgeon? mitigating gender-biased proximities in word embeddings. *Transactions of the Association for Computational Linguistics*, 8:486–503.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2016. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*.

Thomas Manzini, Yao Chong Lim, Yulia Tsvetkov, and A. Black. 2019. Black is to criminal as caucasian is to police: Detecting and removing multiclass bias in word embeddings. In *NAACL-HLT*.

Takeru Miyato, Andrew M Dai, and Ian Goodfellow. 2016. Adversarial training methods for semi-supervised text classification. *arXiv preprint arXiv:1605.07725*.

C. Moss-Racusin, J. F. Dovidio, V. Brescoll, M. Graham, and J. Handelsman. 2012. Science faculty's subtle gender biases favor male students. *Proceedings of the National Academy of Sciences*, 109:16474 – 16479.

Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. 2020. Null it out: Guarding protected attributes by iterative nullspace projection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7237–7256, Online. Association for Computational Linguistics.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1085–1097.

Andras Rozsa, Ethan M Rudd, and Terrance E Boult. 2016. Adversarial diversity and hard positive generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 25–32.

Dongyu Ru, Yating Luo, Lin Qiu, Hao Zhou, Lei Li, Weinan Zhang, and Yong Yu. 2020. Active sentence learning by adversarial uncertainty sampling in discrete space. *arXiv preprint arXiv:2004.08046*.

T. Sun, Andrew Gaut, Shirlyn Tang, Yuxin Huang, M. ElSherief, Jieyu Zhao, Diba Mirza, Elizabeth M. Belding-Royer, Kai-Wei Chang, and William Yang Wang. 2019. Mitigating gender bias in natural language processing: Literature review. *ArXiv*, abs/1906.08976.

Nikhita Vedula, Nedim Lipka, Pranav Maneriker, and Srinivasan Parthasarathy. 2020. Open intent extraction from natural language interactions. In *Proceedings of The Web Conference 2020*, pages 2009–2020.

Tianlu Wang, Xi Victoria Lin, Nazneen Rajani, Vicente Ordonez, and Caimng Xiong. 2020. Double-hard debias: Tailoring word embeddings for gender bias mitigation. *ArXiv*, abs/2005.00965.

Qizhe Xie, Zihang Dai, Yulun Du, E. Hovy, and Graham Neubig. 2017. Controllable invariance through adversarial feature learning. In *NIPS*.

K. Xu, Tongyi Cao, S. Shah, Crystal Maung, and H. Schweitzer. 2017. Cleaning the null space: A privacy mechanism for predictors. In *AAAI*.

B. H. Zhang, B. Lemoine, and Margaret Mitchell. 2018. Mitigating unwanted biases with adversarial learning. *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*.

Huangzhao Zhang, Hao Zhou, Ning Miao, and Lei Li. 2019. Generating fluent adversarial examples for natural languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5564–5569.

Jieyu Zhao, Yichao Zhou, Z. Li, W. Wang, and Kai-Wei Chang. 2018. Learning gender-neutral word embeddings. In *EMNLP*.

Yinhe Zheng, Guanyi Chen, and Minlie Huang. 2020. Out-of-domain detection for natural language understanding in dialog systems. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:1198–1209.