ADEPT: A DEbiasing PrompT Framework

Ke Yang[1], Charles Yu[2], Yi R. Fung[2], Manling Li[2], Heng Ji[2]

[1]Tsinghua University
[2]University of Illinois Urbana-Champaign
yang-k19@mails.tsinghua.edu.cn
{ctyu2,yifung2,manling2,hengji}@illinois.edu

Abstract

Several exisiting approaches have proven that finetuning is an applicable approach for debiasing contextualized word embeddings. Similarly, discrete prompts with semantic meanings have shown to be effective in debiasing tasks. With unfixed mathematical representation at the token level, continuous prompts usually surpass discrete ones at providing a pre-trained language model (PLM) with additional task-specific information. Despite this, relatively few efforts have been made to debias PLMs by prompt tuning with continuous prompts compared to its discrete counterpart. Furthermore, for most debiasing methods that alter a PLM's original parameters, a major problem is the need to not only decrease the bias in the PLM, but also ensure that the PLM does not lose its representation ability. Finetuning methods typically have a hard time maintaining this balance, as they tend to aggressively remove meanings of attribute words (like the words developing our concepts of "male" and "female" for gender), which also leads to an unstable and unpredictable training process. In this paper, we propose **ADEPT**, a method to debias PLMs using prompt tuning while maintaining the delicate balance between removing biases and ensuring representation ability[1]. To achieve this, we propose a new training criterion inspired by manifold learning and equip it with an explicit debiasing term to optimize prompt tuning. In addition, we conduct several experiments with regard to the reliability, quality, and quantity of a previously proposed attribute training corpus in order to obtain a clearer prototype of a certain attribute, which indicates the attribute's position and relative distances to other words on the manifold. We evaluate **ADEPT** on several widely acknowledged debiasing benchmarks and downstream tasks, and find that it achieves competitive results while maintaining (and in some cases even improving) the PLM's representation ability. We further visualize words' correlation before and after debiasing a PLM, and give some possible explanations for the visible effects.

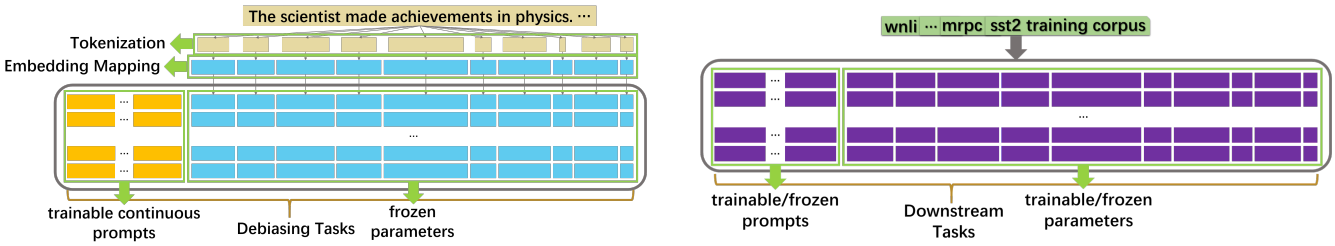
Introduction

Natural Language Processing (NLP) tools are widely used today to perform reasoning and prediction by efficiently condensing the semantic meanings of a token, a sentence or a document. As more powerful NLP models have been developed, many real-world tasks have been automated by the application of these NLP systems. However, a great number of fields and tasks have a high demand for fairness and equality: legal information extraction (Rabelo et al. 2022), resume filtering (Abdollahnejad, Kalman, and Far 2021), and general language assistants (Askell et al. 2021) to name a few. Unfortunately, in the pursuit of the most competitive results, folks often blindly apply PLMs, leading to strong performance with the unseen cost of introducing bias into the process. An ideal NLP tool's decision or choice should not impose harms on a person based on their background (Blodgett et al. 2020), but many studies (Caliskan, Bryson, and Narayanan 2017; Mayfield et al. 2019) have found that biases exist and occur throughout the NLP lifecycle. Thus, it is increasingly important that PLMs can be debiased to enable applications that may be inadvertently influenced by the PLM's implicit stereotypes.

Debiasing, if treated as a special case of downstream tasks, can be tackled through finetuning. Typically, a finetuning debiasing method puts forward specific loss terms to guide a PLM to remove biases in itself (Kaneko and Bollegala 2021). Prompt tuning (Li and Liang 2021; Liu et al. 2021b; Lester, Al-Rfou, and Constant 2021) is one of the more promising methods for transfer learning with large PLMs these days, and its general success (Raffel et al. 2020) suggests applications toward debiasing as well. Prompt tuning, whose role is similar to that of finetuning, refers to freezing all the parameters of the original PLM and only training an additional section of parameters (called a "prompt") for the downstream tasks. Here, a prompt is a set of tokens, often added as a prefix to the input for the task, that act as task-specific complementary information.

All PLM debiasing methods must overcome a major hurdle of "imbalance." Methods that are imbalanced do not adequately balance eliminating biases in a PLM while maintaining its representation ability. Some existing methods are prone to be destructive, whether destructive refers to decreasing a word/sentence embedding's projection on a linear bias subspace (Liang et al. 2020), or refers to completely removing the semantic meanings of attribute words (e.g., man, male; and woman, female) from all neutral words (e.g., engineer, scientist; and teacher, librarian) (Kaneko and Bollegala 2021). If a debiasing framework focuses only on the

[1]The code and data are publicly available at https://github.com/EmpathYang/ADEPT.



(a) While debiasing, ADEPT only trains the prompt parameters and keeps the base model frozen.

(b) When performing downstream tasks, ADEPT conditions either the base model or both the prompt and the base model.

Figure 1: An illustration of how debiasing works using ADEPT and for downstream tasks.

PLM's debiasing task and pays no attention to preserving the model's useful properties, it may destroy the PLM's computational structure and counteract the benefits of pretraining altogether. Although an extreme example, a randomly initialized model is expected to be completely unbiased.

In this paper, we propose **ADEPT** (Figure 1), a debiasing algorithm which implements prompt tuning to debias PLMs and makes the following contributions:

- We are the first to exploit prompt tuning in the debiasing space.
- We introduce a novel debiasing criterion, which often enables the debiased model to perform better than the original one in downstream tasks.
- We show that **ADEPT** is more effective at mitigating biases on a word embedding manifold than other methods which operate on a linear bias subspace.
- We show methods for improving prototypes for contextualized word embeddings that are generated via aggregation.

Our prompt tuning approach has the inherent advantage of saving computing and storage resources. In our experiments, we achieve great results by training prompts with less than 1% the parameters of the PLM as opposed to fine-tuning approaches which train the whole model. Furthermore, because prompt tuning only trains prompt and the PLM's original parameters are not touched during the training process, the base model will maintain its robustness.

Related Work

Debiasing Methods

Word Embeddings Static word embeddings, the foundational building blocks of neural language models, have been a prime target for criticism. In light of artifacts from their training process leading to the encoding of stereotypes, many efforts have been made to mitigate the correlations stored within static embeddings (Mikolov, Yih, and Zweig 2013; Bolukbasi et al. 2016; Caliskan, Bryson, and Narayanan 2017; Mikolov et al. 2013; Manzini et al. 2019). However, most modern PLMs employ contextualized word embeddings, spreading the potentially biased representations of words across various contexts.

Discrete Prompts Solaiman and Dennison (2021) propose PALMS with Values-Targeted Datasets, which finetunes large-scaled PLMs on a predetermined set of social values in order to reduce PLMs' biases and toxicity. Askell et al. (2021) use a hand-designed prompt with more than 4600 solid words as a stronger baseline for helpfulness, harmlessness, and honesty principle for a general language assistant. Schick, Udupa, and Schütze (2021) encourage a model to generate biased text and discard its undesired behaviors with this internal knowledge.

In general, discrete prompts debias PLMs in the form of debiasing descriptions. As crafting discrete prompts manually requires domain knowledge and professional expertise, and we cannot ensure hand-crafted prompts' effectiveness beforehand, we hope to improve debiasing prompts' performance by transforming it to continuous ones which can be optimized with standard techniques like gradient descent.

Finetuning Setting Kaneko and Bollegala (2021) propose a finetuning method of debiasing PLMs. It sets special loss for the debiasing tasks which takes both a PLM's debiasing results and its expressiveness into account. The experiment shows that token-level debiasing across all layers of the PLM produces the best performance. It further conducts experiments on MNLI tasks and finds that the debiased model preserves semantic information. As this work also makes efforts to maintain a PLM's expressiveness while debiasing, we take their debiased model as our baseline.

Prompt Tuning

Prompt usually has two connotations. One is the text with natural semantics, which is fed into the language model together with the original input as additional information. Another is a set of prefixed, continuous trainable numbers post-set into a PLM, which usually do not have semantic meanings. Because this set of continuous numbers have the same functions as the discrete prompt, such as providing the PLM with extra hints for solving a problem, it is also called prompt (or prefix).

Li and Liang (2021), Liu et al. (2021b), and Lester, Al-Rfou, and Constant (2021) propose prompt tuning (or prefix-tuning, p-tuning) as a lightweight alternative to finetuning for performing downstream tasks. This approach conditions a large-scaled PLM by freezing its original parameters and

Algorithm 1: **ADEPT**: a debiasing algorithm for contextualized word embeddings.

Input: a Pre-trained Language Model (PLM)
Output: Φ_{prompt} for debiasing the PLM
ADEPT:

Prepare a PLM M_Θ with parameters Θ.
Suppose a bias has d attributes. Define a neutral word tuple $W^{neutral}$ and attribute word tuples $W^{a(i)} = (w_1^{a(i)}, ..., w_g^{a(i)})$, each with g one-to-one words.
Collect sentences $S^{neutral}$ and $\{S^{a(i)}\}_{i=1}^d$.
Initialize parameters Φ_{prompt}.
for epoch in 1, ..., $epoch_{max}$ **do**
    Calculate prototypes of the neutral words: $E^{neutral} = M_\Theta^{'}(S^{neutral})$, where $M_\Theta^{'} = M_{\Theta \cup \Phi_{prompt}}$.
    Calculate prototypes of attributes: $E^{a(i)} = M_\Theta^{'}(S^{a(i)}), e^{a(i)} = aver(E^{a(i)})$.
    Calculate distances between attribute words and neutral words: $P^{a(i)} = Distance(E^{neutral}|e^{a(i)})$.
    Calculate loss of bias: $L_{bias} = \sum_{i,j \in \{1,...,d\}, i<j} \{JS(P^{a(i)}||P^{a(j)})\}$.
    Calculate loss of representation: $L_{representation} = KL(M_\Theta(S)||M_\Theta^{'}(S))$, where $S = S^{neutral} \cup \{S^{a(i)}\}_{i=1}^d$.
    Calculate the total loss: $L = L_{bias} + \lambda L_{representation}$.
    Compute gradient.
    Update Φ_{prompt}.
end for
return best Φ_{prompt}

optimizing a small continuous task-specific embeddings. Besides saving computing and storage resources, prompt tuning performs even better when the PLM scales up and keeps the PLM's robustness to domain transfer. Our work benefits from these advantages as we debias large PLMs and evaluate their expressiveness on downstream tasks.

Manifold Learning

Manifold learning refers to a series of machine learning methods based on manifold assumption (Melas-Kyriazi 2020). In order to grasp knowledge from data, we need to hypothesize that data has its inborn structure. Manifold assumption indicates that the observed data lie on a low-dimensional manifold embedded in a higher-dimensional space, for example, a Swiss Roll alike data structure in a 3-dimensional data space. t-SNE (Van der Maaten and Hinton 2008), a decomposition method based on manifold assumption, provides excellent visualizations for high-dimensional data that lie on several different, but related, low-dimensional manifolds. For word embeddings with high dimensions, we believe we can better describe its distribution with a manifold than with a linear subspace.

Methodology

Our goal is: given a PLM M_Θ with parameters Θ, find the parameters Φ_{prompt} determining a set of continuous prompts, so that the prompt-tuned model $M_{\Theta \cup \Phi_{prompt}}$ (we will use $M_\Theta^{'}$ for short) has the debiasing effects while maintaining the expressiveness of M_Θ.

We optimize Φ_{prompt} by using the objective function:

$$L = L_{bias} + \lambda L_{representation} \quad (1)$$

where L_{bias} seeks to minimize biases in $M_\Theta^{'}$ whereas $L_{representation}$ caters to the debiased model's expressiveness, and λ is a coefficient to balance the two dependent terms. Our algorithm is summarized in Algorithm 1.

Define Word Tuples and Collect Sentences

We define a neutral word tuple $W^{neutral}$ and several attribute word tuples $W^{a(i)}, i = 1, ..., d$, where the category of bias we are debiasing for contains d different attributes. For example, gender bias may have the attributes "female" and "male," and here $d = 2$ [2]. Words in $W^{neutral}$ are nouns or adjectives that should show no preference for any of the d attributes. For example, "science" and "ambitious," which should not be bound to any attributes, might be in the tuple $W^{neutral}$. $W^{a(i)}$ denotes a tuple of words where each word is associated attribute $a(i)$ and not $a(j)$ for any $j \neq i$. For example, W^{male} might contain the words "uncle" and "masculine" but not contain the word "science" (since that is a neutral word) or the word "parent" (as this is not specific to the "male" attribute). Next, we enforce that each attribute word tuple is indexed by the same (implicit) indexing set of *concepts* and that the word at each index is of the same form. For example, if the (implicit) indexing set is ("parent's sibling", "parent", "sibling") then the male attribute tuple would be ("uncle", "father", "brother") and the female attribute tuple would be ("aunt", "mother", "sister"). For brevity, we use the word "pairwise" to describe this correspondence, although the method can be extended to biases with $d > 2$ as well.

We then collect sentences based on the word tuples. $S^{neutral}$ (or $S^{a(i)}$) denotes sentences that contain at least one word in $W^{neutral}$ (or $W^{a(i)}$, respectively). Instead of creating template-based sentences using the attribute words from $\{W^{neutral}\} \cup \{W^{a(i)}\}_{i=1}^d$, we scrape natural sentences from a corpus (possibly distinct from and/or smaller than the PLM's pretraining corpus) for a diverse word distribution that aligns better with the real-world.

Calculate Prototypes of Neutral Words/Attributes

To get an insight of a model's view on different groups, we seek prototypes of neutral words and attributes. To obtain these prototypes, we extract embeddings for each word. For a word from W^x (x is $neutral$ or $a(i)$ for some i), we fetch the associated sentence from S^x and feed it into $M_\Theta^{'}$. Then, we extract the hidden state for the word from each layer

[2] We hold the opinion that gender identity need not be restricted to the binary choice of male or female. However, for the purposes of experimentation and following prior studies, we adopt this binary setting.

of the forward pass. For PLMs adopting WordPiece embeddings such as BERT (Devlin et al. 2018), if a word has several sub-tokens, we average the sub-tokens' hidden states as the word's hidden state.

For each word tuple's sentences S^x, we extract the set of embeddings E^x. For attribute words, we follow the procedures from Bommasani, Davis, and Cardie (2020) and average the embeddings $E^{a(i)}$ to get a single embedding $e^{a(i)}$ that closer resembles a static embedding as opposed to contextualized embeddings. Under the law of large numbers, we expect this simple linear computation to reduce the context's linear influences on each attribute word. This process can be summarized as:

$$\begin{aligned} E^{neutral} &= M_{\Theta}^{'}(S^{neutral}) \\ E^{a(i)} &= M_{\Theta}^{'}(S^{a(i)}) \end{aligned} \tag{2}$$

$$e^{a(i)} = aver(E^{a(i)}) \tag{3}$$

Thus we take $E^{neutral} = [e_1^{neutral}, e_2^{neutral}, ...]$ as the prototypes of neutral words and $e^{a(i)}$ as the prototype of an attribute.

Define Tuning Loss

We treat word embeddings as being distributed on a manifold and design the loss adhering to the criterion that pairwise attribute words should look alike compared to neutral words on the manifold.

We first design L_{bias} with the intention of pushing pairwise attribute words closer together on the manifold, which corresponds to decreasing biases in a PLM. $p_{n_j|a(i)}$ quantifies the degree to which attribute $a(i)$'s information can be restored from the neutral word n_j in $M_{\Theta}^{'}$:

$$p_{n_j|a(i)} = \frac{exp(-\frac{||e^{a(i)} - e_j^{neutral}||^2}{2\rho^2})}{\sum_{n_k \in W^{neutral}} \{exp(-\frac{||e^{a(i)} - e_k^{neutral}||^2}{2\rho^2})\}} \tag{4}$$

where ρ is a hyperparameter. We can interpret Equation 4 in this way: (1) Let us set a Gaussian distribution with a covariance matrix to be ρ times the identity matrix at the prototype of attribute $a(i)$, which is $e^{a(i)}$. Then the prototype of the neutral word n_j, which is $e_j^{neutral}$, shows up in the distribution with the probability proportional to $exp(-\frac{||e^{a(i)} - e_j^{neutral}||^2}{2\rho^2})$, the numerator. (2) The denominator sums up the probability mentioned above from all $n_k \in W^{neutral}$, and plays a role as the normalization factor. (3) Equation 4 is a formulation that quantifies how much information of $e^{a(i)}$ we can restore from $e_j^{neutral}$. Similar equations have been used in other contexts (Parzen 1962; Hinton and Roweis 2002).

$P^{a(i)}$ denotes our distances from attribute $a(i)$ to all neutral words. $P^{a(i)} = [p_{n_1|a(i)}, p_{n_2|a(i)}, ...]$ means it is a list of values calculated from $e^{a(i)}$ and $E^{neutral}$. Therefore, we summarize it as below:

$$P^{a(i)} = Distance(E^{neutral}|e^{a(i)}) \tag{5}$$

We define L_{bias} as:

$$L_{bias} = \sum_{i,j \in \{1,...,d\}, i<j} \{JS(P^{a(i)}||P^{a(j)})\} \tag{6}$$

where $JS(P^{a(i)}||P^{a(j)})$ is the Jensen-Shannon divergence between distribution $P^{a(i)}$ and distribution $P^{a(j)}$. This loss term is intended to make up the difference between distinct attributes' relative distances to the same group of neutral words (in the form of distribution $P^{a(i)}$ and $P^{a(j)}$) so as to push pairwise attribute words closer.

We then design $L_{representation}$ with the intention of maintaining words' relative distances, which corresponds to maintaining the PLM's representation ability. $p_{w_j|w_i}$ quantifies the degree to which the word w_i's information can be restored from the word w_j in $M_{\Theta}^{'}$. P denotes the matrix of $p_{w_j|w_i}$ where $P_{ij} = p_{w_j|w_i}$. For $q_{w_j|w_i}$ and Q, they denote likewise except that the model is the original one M_{Θ}. $p_{w_j|w_i}$ and $q_{w_j|w_i}$ have the same definition as in Equation 4.

We define $L_{representation}$ as:

$$\begin{aligned} L_{representation} &= KL(Q||P) \\ &= \sum_{i=1}^{||V||} \sum_{j=1}^{||V||} Q_{ij} log_2(\frac{Q_{ij}}{P_{ij}}) \end{aligned} \tag{7}$$

where $||V||$ denotes vocabulary size.

In Algorithm 1, we write $L_{representation}$ as:

$$L_{representation} = KL(M_{\Theta}(S)||M_{\Theta}^{'}(S)) \tag{8}$$

where S denotes the union of $S^{neutral}$ and $\{S^{a(i)}\}_{i=1}^{d}$. Here the $L_{representation}$ aims to keep the PLM's parameters unchanged. Rather than using L_2 norm to gauge how much the outputs of the debiased model has changed as Kaneko and Bollegala (2021) do, we measure the differential between the original model's hidden states and the debiased model's hidden states with KL divergence. $L_{representation}$ in Equation 8 is more time-efficient for training and evaluation tasks than the one in Equation 7, so we adopt it in **ADEPT**.

Improve Prototypes of Attributes

After we confirm that Algorithm 1 works, we make efforts to improve prototypes of attributes $e^{a(i)}$ by adjusting properties of $S^{a(i)}$. We can tell from Equation 3 that $e^{a(i)}$ is a calculated prototype with intuitive correctness. Therefore, we implement experiments on deciding on the desirable properties of $S^{a(i)}$ regarding its reliability, quality and quantity, altering single variable at a time, to check whether $e^{a(i)}$'s expressiveness can be improved with the modified $S^{a(i)}$. The test granularity extends from a single word to the whole attribute.

$S_m^{a(i)}$ denotes a sub-list of $S^{a(i)}$ composed with sentences that contain $w_m^{a(i)}$, where $w_m^{a(i)}$ means the m^{th} item of tuple $W^{a(i)}$. $len(S_m^{a(i)})$ and $len(S^{a(i)})$ denote the length of the lists.

Reliability Here, the experiment is devised to answer: if $len(S_m^{a(i)})$ is less than a threshold, shall we take the word $w_m^{a(i)}$ as a contributing word for constructing $e^{a(i)}$? To satisfy the law of large number, we set the threshold to be 30.

Quality Here, the experiment is devised to answer: if $len(S_m^{a(1)}) \neq len(S_m^{a(2)}) \neq ...$, which is often the case, will this disproportion of pairwise words affect $e^{a(i)}$'s expressiveness? We set $len(S_m^{a(1)}) = len(S_m^{a(2)}) = ..., m \in [1, g]$ and compare the results.

Quantity Here, the experiment is devised to answer: whether for $len(S^{a(i)})$, the larger, the better? We conduct the tests with the $len(S^{a(i)})$ being of a different order of magnitude.

Experiments

Datasets, Benchmarks and Baselines

For the word tuples, we use neutral word lists employed in previous debiasing methods (Kaneko and Bollegala 2021; Caliskan, Bryson, and Narayanan 2017). For the binary gender setting, we use the pairwise attribute words from Zhao et al. (2018) and for the ternary religion setting, we use the attribute triplets from Liang et al. (2020). For the sentences associated with the word tuples, we draw sentences from News-Commentary v15 (Tiedemann 2012) for the gender setting and sentences from BookCorpus (Zhu et al. 2015) and News-Commentary v15 (Tiedemann 2012) for the relgions setting. Since the original BookCorpus is no longer available, we use (lewtun et al. 2022) which is an open source replica. We use this corpus since BookCorpus is part of the corpus BERT is originally trained on. In total, for the gender setting, we draw 20,710 neutral sentences and 44,683 sentences each for the male and female attributes. For the religion setting, we draw 73,438 neutral sentences and 5,972 sentences corresponding to each attribute (Judaism, Christianity, and Islam).

We evaluate gender stereotype scores on SEAT 6, 7, 8 (May et al. 2019) and CrowS-Pairs (Nangia et al. 2020), widely-used benchmarks/metrics designed to evaluate a model's biases toward/against different social groups. We evaluate the debiased models' representation ability on selected GLUE (Wang et al. 2018) tasks, each of which is with little training data: Stanford Sentiment Treebank (SST-2, (Socher et al. 2013)), Microsoft Research Paraphrase Corpus (MRPC, (Dolan and Brockett 2005)) Recognizing Textual Entailment (RTE, (Dagan, Glickman, and Magnini 2005; Haim et al. 2006; Giampiccolo et al. 2007; Bentivogli et al. 2009)) and Winograd Schema Challenge (WNLI, (Levesque, Davis, and Morgenstern 2012)). We further evaluate the comprehensive performance of the debiased model pertaining to its biases and expressiveness on a filtered portion of the StereoSet-Intrasentence data (Nadeem, Bethke, and Reddy 2020), employing 149 test examples for the gender domain and 5,770 test examples overall.

We compare our algorithm with Debiasing Pre-trained Contextualised Embeddings (**DPCE**; (Kaneko and Bollegala 2021)), a similar method that focuses on making the neutral words' embeddings devoid of information in relation to a protected attribute by finetuning the model with the loss term being the sum of the inner product between the attribute words' hidden states and the neutral words' hidden states.

Hyperparameters

We conduct experiments on the BERT-LARGE-UNCASED pre-trained model from HuggingFace (Wolf et al. 2019). By using **ADEPT**, we need only train 1.97M parameters when prompt-tuning with 40 prompt tokens, orders of magnitude smaller than the 335M parameters required for finetuning.

We set λ in Equation 1 to be $\frac{7}{3}$ and ρ in Equation 4 to be 15. We use Adam (Kingma and Ba 2014) to optimize the objective function. During the debiasing process, our learning rate is 5e-5 and our batchsize is 32. Results for **DPCE** are using the hyperparameters originally reported in Kaneko and Bollegala (2021). All the experiments are conducted on two GeForce RTX 3090 GPUs and in a Linux operating system.

Bias Benchmarks

We use three main benchmarks for evaluating performance vis-a-vis bias.

SEAT The Sentence Encoder Association Test (SEAT) (May et al. 2019) extends the Word-Embedding Association Test (WEAT) (Caliskan, Bryson, and Narayanan 2017) to the sentence-level by filling hand-crafted templates with the words in WEAT. In this way, SEAT aims to measure biases in sentence-encoders like ELMo (Peters et al. 2018) and BERT (Devlin et al. 2018) as opposed to only the biases in word embeddings. The SEAT benchmark provides two scores, namely effect size and P-value, where an effect size with smaller absolute value is regarded as a better score for a debiased model.

CrowS-Pairs CrowS-Pairs (Nangia et al. 2020) features pairwise test sentences, differing only in a stereotyped word and an anti-stereotyped word in the same position. This benchmark evaluates whether a PLM will assign a higher probability to a stereotyped sentence than to an anti-stereotyped one where the probability is assigned while attempting to account for differing priors. A ideal model will get the score of 50.

StereoSet StereoSet (Nadeem, Bethke, and Reddy 2020) measures both a PLM's useful semantic information as well as its biases by using cloze tests. Provided a brief context, a PLM must choose its preference from a stereotype, an anti-stereotype, and an unrelated choice. A higher, up to 100, Language Modeling Score (LMS) indicates better expressiveness, and a Stereotype Score (SS) closer to 50 indicates less biases. The Idealized CAT Score (ICAT) is a combined score of LMS and SS with the best score being 100.

Results and Analysis

We evaluate four models on all benchmarks, namely the **original** model (pre-trained with no explicit debiasing), the **DPCE** model, the **ADEPT-finetuning** model finetuned following our debiasing criterion, and the **ADEPT** model (ours). For the CrowS-Pairs and StereoSet experiments, we inherit the classifier from the BERT-LARGE-UNCASED model to predict the masked token, so we perform CrowS-Pairs and StereoSet evaluations on the model with the slightest change. As a result, we choose **ADEPT** model with 500 training steps in these two benchmarks' evaluation. For

	original	DPCE	ADEPT-finetuning	ADEPT	
C6: M/F Names, Career/Family	0.369	0.936	0.328	**0.120**	
C7: M/F Terms, Math/Arts	0.418	-0.812	**-0.270**	-0.571	
C8: M/F Terms, Science/Arts	-0.259	-0.938	-0.140	**0.132**	
CrowS-Pairs: score(S)	55.73	47.71	52.29	**48.85**	
GLUE: SST-2	92.8	92.8	**93.6**	93.3	92.7
GLUE: MRPC	83.1	70.3	83.6	84.6	**85.0**
GLUE: RTE	69.3	61.0	69.0	**69.7**	**69.7**
GLUE: WNLI	53.5	45.1	46.5	47.9	**56.3**
StereoSet(filtered)-gender: LMS	**86.338**	84.420	86.005	84.652	
StereoSet(filtered)-gender: SS	59.657	59.657	57.113	**56.019**	
StereoSet(filtered)-gender: ICAT	69.663	68.115	73.770	**74.462**	
StereoSet(filtered)-overall: LMS	84.162	58.044	**84.424**	83.875	
StereoSet(filtered)-overall: SS	58.243	**51.498**	57.701	55.435	
StereoSet(filtered)-overall: ICAT	70.288	56.305	71.420	**74.759**	

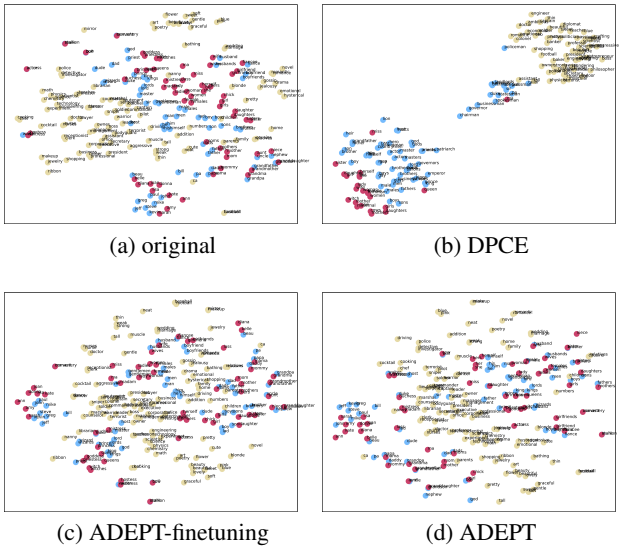
Table 1: Evaluation results on debiasing performance. We test the debiased models on SEAT (from row 1 to 3), CrowS-Pairs (row 4), GLUE (from row 5 to 8) and filtered StereoSet-Intrasentence (from row 9 to 14), with best result in bold. original, the original BERT-LARGE-UNCASED model; DPCE (Kaneko and Bollegala 2021), the baseline model; ADEPT-finetuning, model finetuned with our new debiasing criterion; ADEPT, model tuned with ADEPT. As for downstream tasks in GLUE, we test ADEPT on both fine-tuning the original model only (left) and fine-tuning the model as well as the debiasing prompt (right).

SEAT and GLUE, we use the **ADEPT** model after 10 epochs of training.

Reducing Biases In Table 1, experiments show that **ADEPT** achieves competitive debiasing results, outperforming **DPCE** and mostly obtaining the best scores of the four models on SEAT and CrowS-Pairs. **ADEPT-finetuning**, which shifts **ADEPT** from the prompt-tuning setting to a finetuning setting, is also broadly successful at eliminates biases in PLMs. More trainable parameters notwithstanding, **ADEPT-finetuning** fails to beat **ADEPT**, implying that debiasing does not require a great change to the original model.

Preserving Representation Ability In Table 1, the GLUE tests show that **ADEPT** does not harm the model's representation ability and even improves it in most cases, with increased scores on SST-2, MRPC, and RTE. As shown in Figure 1(b), after being debiased with **ADEPT**, the model can choose from training the base model or both the prompt and the base model when performing downstream tasks, and we test both on selected GLUE tasks. Notably, the trainable prompt parameters account for less than 1/100 of the base model parameters, so additionally training the prompts does not significantly add to the computational burden. We can see enhanced performance in both cases. **ADEPT-finetuning** also manages to outperform the **original** pretrained model on SST-2 and MRPC, although the overall results are less noteworthy.

Visualization and Comprehensive Performance We explore the visible increase in the debiased model's expressiveness by visualizing words' correlation given by the model before and after debiasing, and provide a comprehensive



(a) original (b) DPCE

(c) ADEPT-finetuning (d) ADEPT

Figure 2: Visualized correlation of words in the gender domain. We use t-SNE to plot the figures and set perplexity as 30. We color neutral words beige, male words blue, and female words red.

score on the filtered StereoSet-Intrasentence. For a better prototype of a word, we average the last layer hidden state of the word from 30 different sentences. We plot **ADEPT**'s performance on binary gender debiasing in Figure 2(d) (we also plot the results for ternary religion debiasing in the Appendix). We filter the StereoSet-Intrasentence dataset to only

	LMS	SS	ICAT	score(S)
raw	86.674	62.341	65.282	52.29
reliability	85.975	61.846	65.605	53.05
quality	86.728	62.329	65.343	53.44
quantity-100	86.493	60.857	67.712	53.82
quantity-1000	86.166	61.168	66.920	51.91
quantity-10000	86.753	61.550	66.713	52.29

Table 2: Experiment results on desirable properties of $S^{a(i)}$ as detailed in the section on Improving Prototypes.

keep test examples with the target words "daddy," "ma'am," "groom," "bride," "stepfather," or "stepmother," as these words show up less often in the tuning corpus.

From Figure 2 we can conclude that **ADEPT** succeeds at maintaining words' relative distances, while simultaneously pulling pairwise attribute words closer. In comparison, as shown in Figure 2(b), removing attribute semantic meanings as done in **DPCE** splits neutral words and gender words apart, which actually makes the difference between pairwise gender words negligible compared to their relative distances to the neutral words group. This may account for why some previous debiasing methods see a drastic drop in their model's expressiveness after debiasing whereas **ADEPT** does not. We further plot the evaluation loss in the training process in Appendix and find that the training process of **ADEPT** is smoother than that of **DPCE**.

The filtered StereoSet-Intrasentence result also implies that **ADEPT** is better at keeping useful semantic information when eliminating biases. **ADEPT** achieves the best ICAT score across the evaluated models in the filtered StereoSet-Intrasentence for gender and for overall, with best SS in gender domain and best LMS across domains. The baseline **DPCE** model appears to misunderstand words other than gender words as its LMS declines from 84 to 58 when the StereoSet-Intrasentence extends its examples from gender to other protected groups like race and religion. We note that the SS does not improve as much as the SEAT or CrowS-Pairs scores do. We hypothesize that this is because our training process is more similar to the metrics used for SEAT and CrowS-Pairs, which are calculated across the full sentence, rather than to StereoSet, which computes only for the word.

Experiments for Improving Prototypes of Attributes

We feed all sentences in $S^{a(i)}$ into the PLM, average the hidden states of the attribute words, and get a prototype $e^{a(i)}$ of attribute $a(i)$ on the manifold. As we aim to drive pairwise attribute words closer on the manifold, a prototype has to be clear and precise for generalizing the attribute's concept. Therefore, we perform several experiments adjusting the properties of $S^{a(i)}$ to improve $e^{a(i)}$. **raw** denotes the original $S^{a(i)}$.

Reliability As contextualized word embeddings mix context information into every token's hidden states, for word $w_m^{a(i)}$, we need a myriad of context sentences to construct its prototype. Therefore, we regard $S_m^{a(i)}$ with $len(S_m^{a(i)}) < 30$ as unreliable, and remove them from $S^{a(i)}$.

Quality Pairwise attribute words, like "waiter" and "waitress" for gender, should make equal contributions to the prototype. For a word $w_m^{a(i)}$, if its $S_m^{a(i)}$ makes up most of the $S^{a(i)}$, then the calculated prototype may well be influenced by the word's semantic meaning and leads to ambiguity. Thus, we enforce $len(S_m^{a(1)}) = len(S_m^{a(2)}) = ...$ for all pairwise attribute words.

Quantity A larger corpus indicates more diverse training sentences and attribute words, but is more time-consuming to train. Hence, we test $S^{a(i)}$ with sizes at different orders of magnitude and compare the effects to choose the most desirable corpus size.

We run **ADEPT-finetuning** on the corpora mentioned above, stop the training at 500 steps (an early stage), and evaluate the debiased models on StereoSet-Intrasentence and CrowS-Pairs. Results are listed in Table 2. Data show that setting threshold for $S_m^{a(i)}$ and slicing pairwise $S_m^{a(i)}$ to be of equal size help improve the performance. In our experiments, we filter $S_m^{a(i)}$ if $len(S_m^{a(i)}) < 30$, set $len(S_m^{a(1)}) = len(S_m^{a(2)}) = ...$ and choose **quantity-10000**.

Conclusion

We proposed **ADEPT**, an algorithm that adopts prompt tuning for debiasing and introduces a new debiasing criterion inspired by manifold learning. By using prompt tuning, **ADEPT** consumes less computing and storage resources while preserving the base model's parameters, ensuring the model's robustness for other tasks after debiasing. Using this new debiasing criterion, **ADEPT** obtains competitive scores on bias benchmarks and even improves a PLM's representation ability for downstream tasks. By visualizing the words' correlation before and after the PLM is debiased, we find that **ADEPT** drives pairwise attributes closer on the manifold and keeps words' relative distances. **ADEPT** provides a smoother loss function than previous methods, allowing for better use of optimizations like early stopping. We also establish the standard of evaluating the corpus for building attribute prototypes in the contextualized word embedding setting and refine **ADEPT**'s performance with it. In the future, we will explore time-efficient objective terms for keeping words' relative distances in debiasing and release a new dataset that measures a model's biases and expressiveness comprehensively, free of the need to make predictions about masked tokens.

Ethics Statement

When designing **ADEPT**, we made some assumptions to simplify the complex world model which may lead to some ethical concerns.

We defined bias in this paper as the difference between attribute prototypes relative to neutral words. However, this requires carefully selecting the categories of the attribute based on real-world debiasing demands. Unfortunately, our

construction could be reversed such that the words in the attributes list are made as different in distance from the neutral words as possible, but we expect that this would cause the embeddings to degrade and thus not be effective for intentionally causing harm.

In our paper, we discussed the usage of **ADEPT** on the binary gender setting, which in general is not reflective of the real world, where gender (and other biases) can be far from binary. It is reasonable to have concerns that a binary construction can cause harms to groups not part of the pair. Luckily, all pieces of **ADEPT** are directly extensible to any number of dimensions, allowing for all dimensions to be pushed to cluster together.

Unfortunately, we cannot ensure or contradict causality between bias reduction and discrimination mitigation in PLMs. Goldfarb-Tarrant et al. (2020) makes an effort to deny the causality between bias and discrimination, but it only takes WEAT as the intrinsic task, so more work is needed in this area to ensure that we are indeed reducing harms.

Acknowledgements

We thank the anonymous reviewers' helpful suggestions.

Reference

Abdollahnejad, E.; Kalman, M.; and Far, B. H. 2021. A Deep Learning BERT-Based Approach to Person-Job Fit in Talent Recruitment. In *2021 International Conference on Computational Science and Computational Intelligence (CSCI)*, 98–104.

Askell, A.; Bai, Y.; Chen, A.; Drain, D.; Ganguli, D.; Henighan, T.; Jones, A.; Joseph, N.; Mann, B.; DasSarma, N.; et al. 2021. A general language assistant as a laboratory for alignment. *arXiv preprint arXiv:2112.00861*.

Bentivogli, L.; Clark, P.; Dagan, I.; and Giampiccolo, D. 2009. The Fifth PASCAL Recognizing Textual Entailment Challenge. In *TAC*.

Blodgett, S. L.; Barocas, S.; Daumé III, H.; and Wallach, H. 2020. Language (technology) is power: A critical survey of" bias" in nlp. *arXiv preprint arXiv:2005.14050*.

Bolukbasi, T.; Chang, K.-W.; Zou, J. Y.; Saligrama, V.; and Kalai, A. T. 2016. Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings. In Lee, D.; Sugiyama, M.; Luxburg, U.; Guyon, I.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Bommasani, R.; Davis, K.; and Cardie, C. 2020. Interpreting Pretrained Contextualized Representations via Reductions to Static Embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 4758–4781. Online: Association for Computational Linguistics.

lewtun; richarddwang; lhoestq; and thomwolf. 2022. Datasets: bookcorpus. https://huggingface.co/datasets/bookcorpus. Accessed: 2022-07-07.

Caliskan, A.; Bryson, J. J.; and Narayanan, A. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334): 183–186.

Dagan, I.; Glickman, O.; and Magnini, B. 2005. The pascal recognising textual entailment challenge. In *Machine learning challenges workshop*, 177–190. Springer.

Dai, A. M.; and Le, Q. V. 2015. Semi-supervised sequence learning. *Advances in neural information processing systems*, 28.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dolan, W. B.; and Brockett, C. 2005. Automatically Constructing a Corpus of Sentential Paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Giampiccolo, D.; Magnini, B.; Dagan, I.; and Dolan, W. B. 2007. The third pascal recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*, 1–9.

Goldfarb-Tarrant, S.; Marchant, R.; Sánchez, R. M.; Pandya, M.; and Lopez, A. 2020. Intrinsic bias metrics do not correlate with application bias. *arXiv preprint arXiv:2012.15859*.

Guo, Y.; Yang, Y.; and Abbasi, A. 2022. Auto-Debias: Debiasing Masked Language Models with Automated Biased Prompts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1012–1023. Dublin, Ireland: Association for Computational Linguistics.

Haim, R. B.; Dagan, I.; Dolan, B.; Ferro, L.; Giampiccolo, D.; Magnini, B.; and Szpektor, I. 2006. The second pascal recognising textual entailment challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, volume 7.

Hinton, G. E.; and Roweis, S. 2002. Stochastic neighbor embedding. *Advances in neural information processing systems*, 15.

Kaneko, M.; and Bollegala, D. 2019. Gender-preserving Debiasing for Pre-trained Word Embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 1641–1650. Florence, Italy: Association for Computational Linguistics.

Kaneko, M.; and Bollegala, D. 2021. Debiasing pre-trained contextualised embeddings. *arXiv preprint arXiv:2101.09523*.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lester, B.; Al-Rfou, R.; and Constant, N. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Levesque, H.; Davis, E.; and Morgenstern, L. 2012. The winograd schema challenge. In *Thirteenth international conference on the principles of knowledge representation and reasoning*.

Li, X. L.; and Liang, P. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Liang, P. P.; Li, I. M.; Zheng, E.; Lim, Y. C.; Salakhutdinov,

R.; and Morency, L.-P. 2020. Towards debiasing sentence representations. *arXiv preprint arXiv:2007.08100*.

Liu, X.; Zheng, Y.; Du, Z.; Ding, M.; Qian, Y.; Yang, Z.; and Tang, J. 2021b. GPT understands, too. *arXiv preprint arXiv:2103.10385*.

Liu, X.; Ji, K.; Fu, Y.; Du, Z.; Yang, Z.; and Tang, J. 2021a. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*.

Manzini, T.; Lim, Y. C.; Tsvetkov, Y.; and Black, A. W. 2019. Black is to criminal as caucasian is to police: Detecting and removing multiclass bias in word embeddings. *arXiv preprint arXiv:1904.04047*.

May, C.; Wang, A.; Bordia, S.; Bowman, S. R.; and Rudinger, R. 2019. On measuring social biases in sentence encoders. *arXiv preprint arXiv:1903.10561*.

Mayfield, E.; Madaio, M.; Prabhumoye, S.; Gerritsen, D.; McLaughlin, B.; Dixon-Román, E.; and Black, A. W. 2019. Equity beyond bias in language technologies for education. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, 444–460.

Meade, N.; Poole-Dayan, E.; and Reddy, S. 2021. An empirical survey of the effectiveness of debiasing techniques for pre-trained language models. *arXiv preprint arXiv:2110.08527*.

Melas-Kyriazi, L. 2020. The mathematical foundations of manifold learning. *arXiv preprint arXiv:2011.01307*.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T.; Yih, W.-t.; and Zweig, G. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, 746–751.

Nadeem, M.; Bethke, A.; and Reddy, S. 2020. Stereoset: Measuring stereotypical bias in pretrained language models. *arXiv preprint arXiv:2004.09456*.

Nangia, N.; Vania, C.; Bhalerao, R.; and Bowman, S. R. 2020. CrowS-pairs: A challenge dataset for measuring social biases in masked language models. *arXiv preprint arXiv:2010.00133*.

Parzen, E. 1962. On Estimation of a Probability Density Function and Mode. *The Annals of Mathematical Statistics*, 33(3): 1065–1076.

Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2227–2237. New Orleans, Louisiana: Association for Computational Linguistics.

Rabelo, J.; Goebel, R.; Kim, M.-Y.; Kano, Y.; Yoshioka, M.; and Satoh, K. 2022. Overview and Discussion of the Competition on Legal Information Extraction/Entailment (COLIEE) 2021. *The Review of Socionetwork Strategies*, 16(1): 111–133.

Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I.; et al. 2018. Improving language understanding by generative pre-training.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P. J.; et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140): 1–67.

Schick, T.; Udupa, S.; and Schütze, H. 2021. Self-diagnosis and self-debiasing: A proposal for reducing corpus-based bias in nlp. *Transactions of the Association for Computational Linguistics*, 9: 1408–1424.

Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C. D.; Ng, A.; and Potts, C. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 1631–1642. Seattle, Washington, USA: Association for Computational Linguistics.

Solaiman, I.; and Dennison, C. 2021. Process for adapting language models to society (palms) with values-targeted datasets. *Advances in Neural Information Processing Systems*, 34: 5861–5873.

Tiedemann, J. 2012. Parallel Data, Tools and Interfaces in OPUS. In Chair), N. C. C.; Choukri, K.; Declerck, T.; Dogan, M. U.; Maegaard, B.; Mariani, J.; Odijk, J.; and Piperidis, S., eds., *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*. Istanbul, Turkey: European Language Resources Association (ELRA). ISBN 978-2-9517408-7-7.

Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; and Bowman, S. R. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R. R.; and Le, Q. V. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Zhao, J.; Zhou, Y.; Li, Z.; Wang, W.; and Chang, K.-W. 2018. Learning gender-neutral word embeddings. *arXiv preprint arXiv:1809.01496*.

Zhu, Y.; Kiros, R.; Zemel, R.; Salakhutdinov, R.; Urtasun, R.; Torralba, A.; and Fidler, S. 2015. Aligning Books and Movies: Towards Story-Like Visual Explanations by Watching Movies and Reading Books. In *The IEEE International Conference on Computer Vision (ICCV)*.

Zmigrod, R.; Mielke, S. J.; Wallach, H.; and Cotterell, R. 2019. Counterfactual data augmentation for mitigating gender stereotypes in languages with rich morphology. *arXiv preprint arXiv:1906.04571*.