Debiasing Contextualized Word Embeddings with Prefix-Tuning

Johannes Maurin Voshol
Faculty of Science
University of Antwerp
Antwerp, Belgium
maurin.voshol@gmail.com

Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

1 Introduction

Recently, deep contextualized word representations generated by transformer-based models (BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019), DeBERTa (He et al., 2020), GPT3 (Liu et al., 2021b), T5 (Raffel et al., 2020)) have become the state of the art for solving almost all downstream Natural Language Processing (NLP) tasks. These word representations are often more powerful than the non-contextualized variants (Word2Vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014)). However, both types of embeddings are obtained from models that were trained on a corpus in an unsupervised setting. It is not a surprise that these models reflect the bias in our language and society. These biases often take on the form of negative associations about demographic groups or individuals based on race, gender, age and religion. At the moment, these biased models are widely adopted and make biased decisions, harming these demographic groups. There have been proposed various methods trying to remove the bias in non-contextualized word embeddings, but these debiasing methods do not always generalize to contextualized word embeddings. Moreover, these works often claim that the bias was successfully mitigated, without consideration that other tests may find no reduction in bias. This unmeasured bias can still be just as harmful. Therefore, more research is needed towards debiasing word representations.

[. . .] Retraining a model is expensive, we can we make it more efficient in terms of the number of parameters that must be stored.. [...]

In section 3, a summary is given on how to measure (gender) bias in a language model (LM) and the word representations it generates. To clearly define bias, we will need to shed more light on the relationship between our society, its language and the found bias in the corpora. Along with this, the popular methods to measure bias are described in context to downstream tasks. In section 3, we will conduct a literature review and address some popular methods to mitigate gender bias as well as some of the (niche?) latest approaches and categorize them into four categories. We discuss whether methods can be generalized to all LM's and can be applied to contextual models. Then, future directions are described in Section 4. We explore the possibilities and challenges of combining or adapting these methods to improve efficiency or effectiveness. Here we also take into account other recommendations made in literature.

[..]

All code and bias tests can be found in a github repository[1].

The following contributions are made:

- Combining the field of debiasing word embeddings with prompt engineering.
- Log Probability Bias Score (LPBS) tests (adjectives, stereotypes, occupations)

2 Bias in Language Models

2.1 Representational Harms

As formulated by Crawford (2017), bias in pre-trained language model (PLM)'s can result in two types of harm: (1) Allocative harm, which is the unjust distribution of resources and opportunities (e.g. jobs, loans, insurance and education) as the result of an unfair system. (2) Representational harms, which relate to the incorrect identification of groups or individuals by the system. In the past, researchers have been focusing mainly on the allocative harms in downstream tasks. Improving fairness in these tasks is directly observable and easy to quantify (e.g. job allocation among men and women should be 50/50) compared to mitigation of representational harms, which are harder to formalize. However, Crawford (2017) argues that the representational harms are at the root of all forms of allocative harm. Therefore, to make any meaningful progress to improve fairness in the long term, representational harms should receive more attention. There are five types of representational harms, which we will describe in in relation to a PLM: (1) Stereotyping - reinforced existing societal stereotypes, (2) Recognition - algorithm's inaccuracy in recognition tasks, (3) Denigration - culturally or historically derogatory terms, (4) Under-representation - disproportionately low representation of a specific group and (5) Exnomination - considering one specific group as the default.

However, our literature study shows that most methods are effective at mitigating biases in PLM's, but only to a certain degree. There is no way of guaranteeing that a PLM is completely fair. Moreover, bias is often measured in word representations with intrinsic metrics, but in the end we are still interested in the fairness of a system using the PLM.

"Representation bias arises from how we sample from a population during data collection process", (Mehrabi et al., 2021).

2.2 Bias definition

Researchers need to better articulate the conceptualization of "bias" (Blodgett et al., 2020). We need a better understanding of the relationship between language and hierarchies (social sciences, linguistics). When addressing the harm, explicitly state to which group the model is harmful and in what way. Researchers should be transparent about the limitations of their models and the methods used to mitigate bias.

3 Literature Study

3.1 Methods to fine-tune a model

Language models can be optimized for a specific downstream task by fine-tuning on a task-specific dataset. In addition to this, fine-tuning methods can also be used to debias a model. In this section, we will review the various methods for fine-tuning a model, each with its own trade-offs in terms of performance, efficiency, and the number of parameters that must be stored. This will also provide the foundation for understanding the debiasing methods discussed in Section 3.2 and our proposed method based on Prefix-Tuning (Li and Liang, 2021) in Section 4. For each method, the number of parameters that must be stored, compared to the original model, is displayed as percentages.

Retraining parameters (20-100%). Fine-tuning all the parameters in a pre-trained model is a common approach that often results in the best performance. However, this can be computationally expensive, as it requires training all the parameters in the model. Additionally, storing the model can also be space-intensive due to the large number of parameters. Fine-tuning only the last k layers of

[1] https://github.com/m4urin/prefix-debiasing

the model can reduce computational overhead and storage requirements at the cost of some performance.

Adapters (3-4%). Fine-tuning a pre-trained model can lead to the "forgetting" of useful information due to the overwriting of parameters during training. To address this issue, Houlsby et al. (2019) proposed Adapter-Tuning, a method that involves the insertion of feed-forward networks (adapters) between the layers of the pre-trained model. The adapters are trained while the parameters of the original model are kept fixed. This approach has been shown to improve performance while significantly reducing the storage requirements for the model (to 3-4% of the size of the original model).

Prompt-engineering (0%). In recent years, large language model (LLM)'s have become state of the art for most downstream tasks. A drawback of these models is that as they become larger, it will be more expensive to retrain and store them. However, the capabilities of these pre-trained models are powerful enough to perform well in a zero-shot setting using prompts (Wei et al., 2021). Therefore, research shifted from transfer-learning (first pretraining, then fine-tune) to zero-shot approaches, where the internal knowledge of the model can be harnessed to perform well on many downstream tasks. The main strength lies is the fact that the model's parameters do not have to be retrained in any way. The field of prompt-engineering aims to augment the input text to guide a model to even better predictions. In this case, non of the parameters have to be retrained and only the prompts need to be stored. There are several approaches to find prompts that fit the predictions for a given dataset best. Naively, we can express the prompt in our own language and generate templates by hand, selecting the one that yields the best performance.

For example, a masked language model can perform sentiment analysis using the following template: "The sentence '`[Z]`' is `[MASK]`". Here `[Z]` is the original input. The word probabilities for the masked token can be inspected to make a prediction for positive sentiment:

$$P(\texttt{[MASK]}\text{='positive'}) > P(\texttt{[MASK]}\text{='negative'})$$

In another example, when using a PLM in an autoregressive setting (predicting the next word), a summarizer can be implemented using only prompts. When a model has been pre-trained on a large dataset, it might have seen enough examples of summaries as part of a TL;DR (too long; didn't read). We simply append "`TL;DR:`" to the original input and let the auto-regressive model write a summary.

Finding the best prompt is not a trivial task and there are many methods to improve prompts to get better predictions (Liu et al., 2021a). One method to find the best prompt for a given task is AutoPrompt, outlined by Shin et al. (2020). Here, they use a gradient-based search to find a series of tokens (prompt) drawn from the vocabulary that optimizes the accuracy for a task. However, these prompts might find incomprehensible prompts. For example, for a certain model, sentiment analysis might be performed using the prompt: "`atmosphere alot dialogue Clone totally`".

Prefix-Tuning (0.1%). AutoPrompt limits the search space to discrete tokens, or words, that are present in a predefined vocabulary. Prefix-tuning (Li and Liang, 2021) extends this approach by searching a continuous space. Normally, words are first converted to static embeddings. However, in this method, a set of new embeddings can directly be trained to form the prompt. This allows for a more flexible and expressive generation of prompts, as the continuous space of embeddings allows for a greater degree of variation compared to the discrete vocabulary used in AutoPrompt. The effectiveness of Prefix-Tuning on various natural language generation tasks was demonstrated, showing that it obtained comparable performance in the full data setting, outperforms fine-tuning in low-data settings, and extrapolates better to examples with topics unseen during training, while only learning 0.1% of the parameters (Li and Liang, 2021).

3.2 Bias Mitigation

Methods to mitigate bias in PLM's are often categorized as pre-, in- and post-processing methods. Here, pre-processing is based on the manipulation of data before starting the training procedure of the model. Methods that alter the algorithm in its entirety or add additional constraints during the training of the model can be considered in-processing methods. Lastly, post-processing methods can be applied to remove bias in word representations from the finalized model. Note that the latter does not require retraining the PLM. However, for this literature study, we will use the following categories, as the most popular methods can be better grouped into four types of algorithms.

3.2.1 Data manipulation

The observed bias in a language model is merely the reflection of the bias found in the data on which the model was trained, which often originates from real-world observations. The hypothesis is that, by making the data fairer, the language model will become fairer as well. However, it is impossible to acquire perfectly fair data. Therefore, research has been done to debias training corpora that fit our expectations of fairness.

Counterfactual Data Augmentation (CDA). It might prove sufficient to (re)train the model on an augmented version of the corpus in which we artificially add more examples, in this case counterfactual data. This will rebalance the problematic associations between words. It not only reduces the bias in the word representations, but it might also make them more useful in downstream tasks as they have been trained on more positive samples (needs source). Using augmented data, we move away from post-hoc approaches that remove associations (e.g., nurses should not be associated with the female gender) to approaches that balance associations by adding positive examples (e.g., a nurse is just as often a woman as a man). This approach works for both contextualized- and static word embeddings, since they are trained on the same data.

Zhao et al. (2018a) augment the data by duplicating the dataset with the genders swapped and names anonymized. For this they use a word list with gendered pairs. The names present in the corpus are anonymized. However, generating sentences with genders swapped from a corpus is a non-trivial task; (1) it relies on an (incomplete) world list, (2) grammatical errors may occur which will be propagated through the model, (3) it might create nonsensical sentences (e.g. "he gave birth"), (4) as the dataset is duplicated, the training time will increase by a factor of two and (5) in the case of re-training, we need access to the original corpus to generate counterfactuals for the stereotypes the model was trained on. These challenges will be significantly more difficult when dealing with multi-class biases, such as race, religion or age.

Counterfactual Data Substitution (CDS). Maudslay et al. (2019) show that the size of the corpus does not have to be increased, as swapping gender only 50% of the time is just as effective. They apply grammar interventions (Lu et al., 2020) using coreference resolution to generate grammatically correct sentences. This is more difficult to achieve in morphologically rich languages that have many masculine- and feminine-inflected words (e.g., Spanish, Hebrew). Therefore, Zmigrod et al. (2019) investigate the use of a Markov random field inference to alter the grammatical gender of nouns in sentences.

Bias fine-tuning. Kaushik et al. (2019) show the strength of counterfactuals in a downstream task like sentiment analysis. They do this by designing a human-in-the-loop system that alters sentences minimally to change the classification label. However, we do not always have the resources to create a debiased dataset for a downstream task. Another approach by Park et al. (2018) is to first fine-tune on an unbiased dataset of a related task, then fine-tune on your task specific biased dataset. The resulting model contains lower bias, but it is still less effective than CDA. However, bias fine-tuning does not alter the representations produced by

the PLM.

3.2.2 Post-hoc methods

Post-hoc methods try to mitigate the bias in word representations after the PLM has finished training. Advantages of these types of methods are that they are - in comparison to the other methods - relatively cheap in computation, and the parameters of the PLM do not have to be trained again. However, the model itself may still be biased and encode more information into the word representations indirectly related to the bias.

Neutralization and Equalization. Bolukbasi et al. (2016) was the first to calculate a gender direction from male and female word pairs. Words vectors that are supposed to be gender-neutral (e.g., occupations) are projected to the orthogonal subspace as to remove the gender information from the word representations (neutralization) and are put to equidistant to all pairs of gendered words (equalization). This was a good first step in the direction of having fair word representations. However, research showed that capturing the gender subspace and debiasing word representations with it is a non-trivial task. Gonen and Goldberg (2019) showed that the removal of a gender subspace merely hides the bias and that it can be easily recovered. Clustering of the 1,000 most biased words using t-SNE before and after debiasing using the method of Bolukbasi et al. (2016) showed almost two identical results.

RBA. Zhao et al. (2017) proposed constraining the predictions made by the model to match the distributions of the training data. This helps when a model magnifies a stereotype. Debiasing happens at the inference time, no need to retrain the model. TODO

INLP. Ravfogel et al. (2020) improve on neutralization and equalization by proposing INLP (Iterative Null-space Projection), where multiple linear classifiers are trained to find the bias subspaces of specific properties. Then, the word representations are projected onto the null-spaces (orthogonal subspaces) of these classifiers, such that the trained classifier cannot correctly classify word representations. Although this method was applied to contextualized word representations generated by BERT, it does not consider the unlimited number of representations that a contextualized language model can produce for a word. This work was later extended by Ravfogel et al. (2022), by proposing the use of an adversarial to learn the subspace dimensions in a minimax game. Here, the adversarial is limited to a fixed-rank orthogonal projection, as to not remove too much information.

SENT-DEBIAS. Liang et al. (2020) propose a contextualized version of the method of Bolukbasi et al. (2016). Whereas static embeddings do never change after training, contextualized representations can take on many forms. It is therefore necessary to approximate the embedding space of a word (in this case a sentence representation), which is a non-trivial task. Thus, all words belonging to the bias classes are contextualized first by inserting them into templates. This generates many sentence-level representations for each word. Subsequently, bias subspaces are obtained by applying PCA (Principal Component Analysis) to all contextualized sentences. Finally, the sentences are projected to these subspaces.

FairFil. Instead of removing information from the representations, Cheng et al. (2021) propose to append a filter to the model that transforms the representations to debiased versions. First, they augment a corpus through CDA (see related work). The assumption is that the original and augmented sentences semantically mean the same thing. Therefore, both debiased representations should be similar. To train the filter, contrastive learning is applied.

3.2.3 Constrained optimization

Dev et al. (2020) state that one or more linear projections may be considered too aggressive, as valuable information may also be erased from word representations as well (e.g., the association of the word 'birth' with the female gender). Instead of removing concepts

altogether through projection, debiased word representations may also be learned under optimization constraints. The disadvantage of these types of methods is that it is more expensive in computation, as the parameters of the PLM must be retrained again. However, this can vary much between methods and language models. For instance, retraining static embeddings requires significantly less computation compared to transformer-based models.

Gender neutral features. To create gender neutral versions of GloVe embeddings, Zhao et al. (2018b) constrain the learning process by forcing gender information to be encoded in one or more features, freeing the other features from gender information. These gender features can be omitted later if required. To our best knowledge, this method had not been applied to contextualized word representations.

Orthogonal training. Kaneko and Bollegala (2019) append an autoencoder to the model. A classifier is trained to predict gender information from encoded words that must preserve their gender information, while it should not be able to predict gender information in encoded words that are considered neutral- and stereotypical. Moreover, a gender direction is calculated and all neutral- and stereotypical representations are trained to be orthogonal to this gender direction. This is achieved by minimizing the inner product of the representations with the gender direction, which we will address now as 'orthogonal training'. Dev et al. (2020) propose a similar method that rotates target concepts (e.g. occupations) orthogonal to the gender concept during a fine-tuning step, which minimizes similarity between concepts. The concept of orthogonal training was later extended by Kaneko and Bollegala (2019) to fit contextualized language models. Instead of having one representation of a word, there are now many representations of different contexts. Instead of calculating one linear gender subspace using representations of different words, they move to having a linear subspace for each word. Subsequently, the target (e.g., occupations) words are trained to be orthogonal to all other gender words individually.

Dropout regularization. Bias can also be mitigated using dropout regularization. Webster et al. (2020) investigate whether increasing the dropout probabilities for the activation functions and attention mechanisms in BERT and GTP-2 prevent undesirable associations between words. The assumption can be made that increasing the dropout will impact language modeling and downstream tasks. However, Meade et al. (2021) show that the word representations are not damaged to such a critical extent that the model cannot perform on downstream tasks. Moreover, they argue that a fine-tuning step helps the model relearn essential information, even if it was removed during debiasing.

Adversarial Learning. Zhang et al. (2018) propose to use an adversary network, where the generator (the language model) learns to embed words with respect to a gender and prevents the discriminator from identifying the gender. (The discriminator should not be able to predict protected properties, but that does not work well in a contextual setting (???). For example in "She is a nurse", the static embedding of the token of 'nurse' should not display any gender information, whereas the contextualized token in the sentence "She is a nurse" should contain gender information?) The idea of using an adversarial is also used in INLP, but instead of the generator learning to hide the information, the adversarial classification is countered by storing a linear transformation.

ADELE. With *sustainable modular debiasing of language models* (ADELE), Lauscher et al. (2021) apply the adapter technique by injecting adapter modules into the original PLM layers and updates only the adapters through training on a counterfactually augmented corpus (CDA). It is shown to be effective in bias mitigation on several intrinsic and extrinsic benchmarks for BERT and preserves fairness even after large-scale downstream training.

3.2.4 In-context learning

As shown in Section 3.2.2, the internal knowledge of the model was also used to calculate a gender subspace. However, recent developments in zero-shot learning may help us propose better solutions to measure and mitigate biases. It could either be used to create debiased representations directly, or to acquire data from the model itself that can be used in a debiasing fine-tuning step. Therefore, the following methods may also be categorized as either post-hoc or constrained optimization. (? Delobelle et al. (2021), If bias can be measured best in an extrinsic task, what if all downstream tasks use zero-shot learning, making all tasks intrinsic?)

Self-debias. Schick et al. (2021) show that language models can recognize their biases when given a textual description of the undesired behavior. When a language model is used in an autoregressive context, this ability to self-diagnose can be used to reduce the probability of producing biased text. The sentence of interest is inputted twice into the model. For one of the inputs, the model is prompted (preceded) with a text of the undesirable behavior, such as "The following sentence discriminates against people because of their gender". This produces two different probability distributions for the next token, which can be used to suppress biased tokens. However, this post-hoc approach does not change the word representations of the model and can only be used in autoregressive tasks (e.g., Masked LM, Seq2seq).

Auto-Debias. Debiasing results are often subject to the quality of templates and word lists that we choose. As Schick et al. (2021) showed, the model can be elicited to produce biased text with the use of prompting. Guo et al. (2022) use beam search to generate biased prompts. Normally, this method is used to find the most likely sequence output with use of a limited number of branches, but in this setting, it is used to produce sentences that are most biased. When a biased prompt is found for both genders, a fine-tuning step is applied to minimize disagreement between these prompts (need to look at the paper again to check if this is correct).

3.3 Measuring Bias

Analogies. The presence of social biases in the non-contextualized word embeddings was first shown by Bolukbasi et al. (2016) with use of analogies. The difference in the representation of the words "man" and "woman" in the embedding space is approximately the same as the words "king" and "queen". Unfortunately, this is also true for stereotypical words such as "computer programmer" and "homemaker". These properties have lead to various methods to debias word representations, as it is fairly easy to manipulate them with linear transformations. However, analogies are not an accurate tool to measure bias, and can give a distorted view of bias. Nissim et al. (2020) discover that..

Association tests. Most of the intrinsic bias measures are based on the Implicit Association Test (IAT), a psychological measure that is used to assess people's unconscious biases. Caliskan et al. (2017) develop a statistical test, the Word Embedding Association Test (WEAT), analogous to the IAT. It measures the association between sets of target words and attribute words and can be used to investigate the presence of biases in word embeddings. The null hypothesis is that there is no significant association between the target and attribute words. There are many method that extend this method to work with contextualized language models such as the Sentence Encoder Association Test (SEAT) (May et al., 2019), the Mean Average Cosine similarity (MAC) (Manzini et al., 2019) and the Contextualized Embedding Association Test (CEAT) (Guo and Caliskan, 2021).

Coreference resolution. Coreference resolution is a task in which a model must identify expressions that refer to the same entity in a text. To evaluate the bias in a model, it is possible to compare the model's predictions for certain expressions associated gender labeled entities. However, there may be subjectivity

in the choice of templates and scoring functions used, which could affect the results. It is important to consider these limitations when interpreting the results of these type of tests.

There are several extrinsic measures that use this approach, including WinoBias (Zhao et al., 2018a), WinoGender (Rudinger et al., 2018) and Bias-in-Bios (De-Arteaga et al., 2019). These tests can be performed on existing coreference resolution systems. Webster et al. (2020) propose Discovery of Correlations (DisCo), in which a masked language classifier (from pre-training) is used to produce probability scores for gender labeled words. Instead of comparing the coreference prediction for two different gendered words, we directly inspect the word probabilities for a masked word. For this, they use a threshold to label an example as biased.

Given the template "[TARGET] is a [ATTRIBUTE]", where [TARGET] is being masked, there is a high likelihood of a gendered word being biased towards a chosen attribute. However, masking both words might also results in a difference between genders. This suggests that the model exhibits a general preference for a specific gender. Kurita et al. (2019) propose a method to adjust the probabilities using a prior score, as shown in Equation 1, to obtain an Increased Log Probability Score (ILPS).

$$p_{\text{tgt}} = P(\text{[MASK]=[TARGET]} \mid \text{"[MASK] is a [ATTRIBUTE]"})$$

$$p_{\text{prior}} = P(\text{[MASK]=[TARGET]} \mid \text{"[MASK] is a [MASK}_2\text{]"})$$

$$\text{ILPS} = \log \frac{p_{\text{tgt}}}{p_{\text{prior}}} \tag{1}$$

The difference in scores between TARGET words of different genders, denoted with m and f, can then be used to calculate a bias measure for a specific attribute word, referred to as the Log Probability Bias Score (LPBS). We use the logarithm subtraction rule as shown in Equation 2. This metric can be used to determine scores for various combinations of template, attribute, and target words, which can then be combined to obtain a final score indicating the overall bias present in the model. In the results section, the LPBS is typically presented as the average score with the standard deviation, which indicates the variance in bias within the model.

$$\begin{aligned} \text{LPBS} &= |\text{ILPS}^{(m)} - \text{ILPS}^{(f)}| \\ &= |\log \frac{p_{\text{tgt}}^{(m)}\, p_{\text{prior}}^{(f)}}{p_{\text{prior}}^{(m)}\, p_{\text{tgt}}^{(f)}}| \end{aligned} \tag{2}$$

Visualization. Dimensionality reduction techniques, such as PCA and t-SNE, can be used to visualize the biases present in word embeddings. These methods allow for the representation of high-dimensional data in a lower-dimensional space and can therefore be used to show how different words are clustered. This can help us identify new (indirect) biases present in the model for further analysis and action. However, these methods do not provide a complete or fully accurate representation of the data, as some information may be lost during the reduction process.

3.4 Challenges

Choosing lenses. Evaluation metrics, such as those outlined in Section 3.3, can be used to identify and quantify biases present in a language model. However, many of these metrics rely on word lists with human-defined attributes and stereotypes, which do not capture all forms of bias present in the model (Sedoc and Ungar, 2019). Automated methods use clustering techniques like PCA to identify attributes or stereotypes, but still do not guarantee to capture the complete bias. Therefore, the results of these metrics highly depend on the word lists and templates (Delobelle et al., 2021). While these methods can help to measure the presence of bias in a model, it is not possible to definitely prove the absence of bias. At best, these metrics can demonstrate a reduction in bias for a specific set of words or sentences.

Orgad and Belinkov (2022) find that datasets and metrics are often coupled, in which an

evaluation method is only applied to the corresponding dataset. Results may change significantly when methods are applied to other datasets (showing bias for the one, but not the others). This hinders the ability to obtain reliable conclusions in NLP bias research. Decoupling metrics from datasets can result in more stable evaluations.

Another challenge is that some debiasing methods may result in a trade-off between reducing bias and maintaining model quality. This is particularly evident in intrinsic evaluation methods that compare the prediction scores for biased and unbiased instances. Using StereoSet (Nadeem et al., 2020), a fair model should predict similar scores for stereotypical and anti-stereotypical entailments. In this case, a random model would obtain a perfect score. Therefore, it is important to carefully evaluate the quality of the model in addition to the bias scores to ensure that the debiasing method has not significantly compromised the model's language modeling capabilities.

Intrinsic vs. Extrinsic. Delobelle et al. (2021) find that aspects of intrinsic fairness metrics are incompatible when choosing different templates and embeddings. It is true that intrinsic biases in a language model can contribute to extrinsic biases. However, measures do not show a correlation with unfair allocations in downstream tasks. Therefore, it is advised to use a mix of intrinsic metrics that don't use embeddings directly and extrinsic metrics. Orgad and Belinkov (2022) find that in most studies, only a few extrinsic methods are measured, although more can be measured.

- Cao et al. (2022): "First, we cannot assume that an improvement in language model fairness will fix bias in downstream systems. Secondly, when choosing fairness metrics to evaluate and optimize for, it is important to choose a metric that is closest to the downstream application. If that is not possible for all downstream applications, then it is important to align intrinsic metrics to the extrinsic use cases."

Fair training data. As language models tend to magnify stereotypes and other biases, it is important to have fair and equally distributed data among demographic groups. the data to represent society as perfectly fair

Crawford (2017): E.g. if we have a dataset with 80% male sentences and 20% female, than words like engineer or football are occurring more often. This is still the case after balancing the dataset using gender swapping. This still gives the idea that there are more engineers than nurses. In other words, the data is still unbalanced.

In the case that the original corpus is not available, a model can be debiased with an external corpus related to the task. However, the quality of this external corpus is important. Different corpora have various effects on the debiasing results, some do mitigate the bias, whereas others introduce new biases (Guo et al., 2022).

Multilingualism and multiculturalism. The ability to use multiple languages and the presence of multiple cultures present significant challenges for NLP systems. To handle multiple languages, a model must learn the unique characteristics of each language, including grammar, syntax, vocabulary. The model must also learn the cultural conventions of a language, such as the semantics, norms and values. Even within the same language, there are many variations in language among different social- and demographic groups. However, the availability of high-quality, diverse datasets for a wide range of languages and cultures is limited, which can make it difficult for a model to learn these characteristics effectively. As a result, models are often trained on data that is unbalanced and not representative of the social and demographic groups that end up using the system.

Trying to debias models across multiple languages and cultures brings a new set of challenges with it. We need domain experts to identify and address sources of bias that may be present in a specific language or culture, as well as design and implement debiasing techniques and evaluations that are appropriate for

that specific language and culture. For example, Counterfactual Data Augmentation (CDA) may be more difficult in gender-inflected languages as many words in a sentence must be swapped to their gendered counterpart while maintaining correctness in grammar. Addressing these challenges can not only help improve the fairness of NLP systems, but also deepen our understanding of the complex nature of social biases within languages and cultures.

Changing requirements. TODO Since the current unwanted biases are the result of the current cultural and political views, ad-hoc methods might be the solution to quickly fix new unwanted biases in a LLM.

llms can reason about their own biases, so we might end up in a scenario where a fair downstream task might be enough to produce a fair NLP system.

Multi-class Bias. TODO

- How do we construct counterfactuals for other bias classes, such as race, ethnicity, age etc.?
- D-tuples such (Manzini et al., 2019).

Templates for occupations and gendered words

Language models at scale

- With the rise of large language models (gpt3, bloom, ChatGPT) it can be preferred to apply post-hoc methods as retraining parameters is very expensive.
- Tal et al. (2022): "Compared to smaller PLM's, LLM's have higher bias when using prompts (intrinsic), but make less gender errors in WinoGender (extrinsic). We measure bias in three masked language model families (RoBERTa, DeBERTa, and T5) in two setups: directly using prompt based method, and using a downstream task (Winogender). We find on the one hand that larger models receive higher bias scores on the former task, but when evaluated on the latter, they make fewer gender errors."
- (Basta et al., 2019): When classifying grammatical role, BERT doesn't care about word order... except when it matters

Geometry More auxiliary tasks might capture other semantic and syntactic properties / features, but there is less room for other properties. Because of this, it performs worse on downstream tasks.

Does removing gender subspaces make the embeddings more isotropic?

Properties of contextualized word representations

- A word has an endless number of representations.
- Bias may be nonlinear
- Hasan and Curry (2017): Embeddings have a manifold space, propose Manifold Dimensionality Retention. Does orthogonal training still work in a manifold space?
- Ethayarajh (2019): Embeddings are not isotropic, live in a very small cone in the embedding space
- Zhou et al. (2022): Introduce noise data points with contrastive learning to mitigate the anisotropy problem. (This might also reduce bias?)
- Basta et al. (2019): Further describes the difference in bias between contextualized and non-contextualized word embeddings

Linear Bias Subspace Hypothesis.

- Methods assume Linear bias subspace.
- Vargas and Cotterell (2020): Exploring the Linear Subspace Hypothesis in Gender Bias Mitigation

As discussed in Section x, Gonen and Goldberg (2019) show using tSNE that simply projecting the words on a gender axis is not sufficient to remove gender bias.

First off, most post hoc methods assume a linear gender subspace, even Kaneko does define the subspace by averaging embeddings of gendered words. Then all these methods analyze their method using some form of WEAT, which I feel like always does well when embeddings are trained/transformed to be orthogonal. There are also papers assuming non-linear relationships and because of this do not approximate the gender subspace in a linear way, but in a non-linear way. There is another method that uses manifold learning to bring the bias subspace back to a linear space. But most conclude that there is still bias 'encoded in another way' (Manifold Dimensionality Retention).

Methods can also be categorized by the assumptions that are being made when proposing a debiasing method. For example, the assumption that the concept of gender and the associations between words in the embedding space are linear or nonlinear.

Non-linear sub-spaces. TODO A key drawback of this approach is that it relies on an intuitive selection of a few (or a single) gender directions. As aforementioned, finding the subspace(s) of a concept is non-trivial, let alone describing multiple concepts such as race, age or religion.

It can be described as one or more linear subspaces, but it seems impossible to capture the entire gender subspace.

What are the implications of a non-linear bias subspace?

Indirect Bias.

- Debiased occupations still cluster together.
- Then there is also the problem of having indirect bias, where words that were close to each other in the original embedding space ('homemaker' and 'nanny') are still close to each other in the new embedding space after debiasing.
- Gonen and Goldberg (2019): Post-hoc subspace removal merely hides the bias but does not remove it, as most of the bias can be recovered

Model	Type	Number of parameters
DistilBERT	finetune	66985530
	prefix	67584
BERT	finetune	109514298
	prefix	141312
RoBERTa	finetune	124697433
	prefix	141312

Table 1: Parameters to store for prefixes are 0.1% of the original model's size ($N = 8$).

Gonen and Goldberg (2019) found that post-hoc methods based on linear bias subspace removal only hides the bias but does not remove it, as most of the bias can be recovered. This suggests that debiasing methods may not be sufficient on their own to address the problem of bias in word embeddings and that additional measures may be needed.

Du and Joseph (2020) addresses residual bias (indirect bias) where a concept can still be clustered together in the embedding space. This method reduces gender bias significantly when evaluated with intrinsic tests, but does not so in downstream tasks. It means that even after mitigating indirect bias, embeddings still encode gender bias in 'different ways'.

4 Prefix-Tuning to debias word embeddings

There are various debiasing methods for language models that require fine-tuning, such as orthogonal training and adapters (as discussed in Section 3.2). In an effort to develop a debiasing technique that extends on other fine-tuning approaches, we propose the use of prefix-tuning as a potential method for generating unbiased word embeddings. The assumption that models can recognize their own biases to some extent when given a prompt (Schick et al., 2021) serves as the basis for this approach. We have seen in Section 3.1 that a prompt can drastically change the distribution of word predictions made by a linear classifier. Therefore, it is likely that prompts can also influence the word embeddings in a meaningful way. First, we introduce how a prefix P is

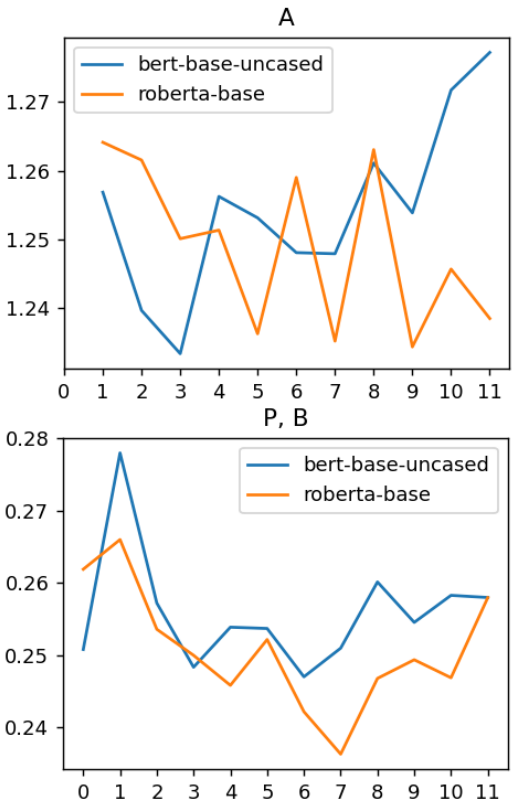

Figure 1: Regularization loss in each layer.

concatenated to a sequence of input tokens x.

$$x \in \mathbb{R}^{|x| \times D} \quad \text{(input)} \quad (3)$$

$$P \in \mathbb{R}^{N \times D} \quad \text{(prefix)} \quad (4)$$

$$[P,\ x] \in \mathbb{R}^{(N+|x|) \times D} \quad \text{(concatenation)} \quad (5)$$

The input to the model, x, is obtained by tokenizing a sentence and has a size of $|x|$, and the prefix consists of N tokens, each with model-specific embedding dimensions of D. Therefore, we are learning N distinct token embeddings that make up the prefix. The notation $[\,\cdot\,,\,\cdot\,]$ refers to the concatenation of the tokens within a sentence, allowing for the individual addressing of the embeddings of the prefix and input sentence. The embeddings for an input sentence are generated by the function E using the language model LM_θ. These embeddings are represented by $p^{(x)}$ and $h^{(x)}$, respectively, with the superscript (x) indicating that the embeddings are influenced by the input sentence x.

$$[p^{(x)},\ h^{(x)}] = E([P_\theta,\ x]; LM_\theta) \quad (6)$$

4.1 Parameterization

In order to improve the performance of the debiasing process, we introduce A and B, which transform the embeddings of the prefix in intermediate layers of the language model. These parameters are used as coefficients and biases, rather than using a $D \times D$ matrix transformation in order to keep the number of parameters small.

$$A \in \mathbb{R}^{\ell-1 \times N \times D} \quad (7)$$

$$B \in \mathbb{R}^{\ell-1 \times N \times D} \quad (8)$$

For each layer i of the language model, with ℓ total layers, we update the prefix embeddings using the function f_i, which applies the transformation defined by A and B through the Hadamard product. The resulting embeddings for layer i are represented by E_i. In the case of transformer-based models, this transformation occurs within a transformer block. Overall, this approach increases the number of parameters by a factor of 2ℓ.

$$[p_0,\ h_0^{(x)}] = [P,\ x] \quad (9)$$

$$[p_{i+1}^{(x)}, h_{i+1}^{(x)}] = E_i([f_i(p_i),\ h_i^{(x)}]; LM_\theta) \quad (10)$$

$$f_i(v) = v \odot A_i + B_i \quad (11)$$

By using regularization terms, as defined in Equation 12, we can regularize the parameters in A and B. If the debiasing process does not require the use of A and B, the regularization loss, $L_{\text{reg}}^{(\text{prefix})}$, will approach 0. This means that the embedded features are being multiplied by 1 and no bias being added, effectively resulting in the same function as described in Equation 6 as f_i acts as an identity function.

$$L_{\text{reg}}^{(\text{prefix})} = \sum (1 - A)^2 + \sum B^2 \quad (12)$$

Furthermore, we can analyze the effectiveness of debiasing in different layers by examining the value of $L_{\text{reg}}^{(\text{prefix})}$. This allows us to determine in which layers the most bias can be effectively mitigated.

4.2 Orthogonal training

The debiasing method proposed by Kaneko and Bollegala (2021) involves learning new representations for target words in the vocabulary $\mathcal{V}_t$ such that these words are orthogonal to attribute words in the vocabulary $\mathcal{V}_a$ with respect to their embeddings in a given sentence x. The embedding of a word w at the i-th layer in x is denoted as $h_{i,w}^{(x)} \in \mathbb{R}^D$. The new representations are learned by considering all sentences containing a target word w, which are retrieved using the function $\Omega(w)$. The representations for attribute words $v_i(a)$ are pre-computed.

$$L_i = \sum_{t \in \mathcal{V}_t} \sum_{x \in \Omega(t)} \sum_{a \in \mathcal{V}_a} (\boldsymbol{v}_i(a)^\top h_{i,t}^{(x)}) \quad (13)$$

$$\boldsymbol{v}_i(a) = \frac{1}{|\Omega(a)|} \sum_{x \in \Omega(a)} h_{i,a}^{(x)} \quad (14)$$

To prevent the model from simply rotating all word embeddings away from the attribute word representations $v_i(a)$, including the attribute words themselves, an additional regularization loss is introduced. This loss is represented by Equation 3, in which the hidden state $\mathbf{h}$ represents the original model's embeddings and remains fixed during training.

$$L_{\text{reg}}^{(\text{debias})} = \sum_{x \in \mathcal{A}} \sum_{w \in x} \sum_{i=1}^{\ell} ||h_{i,w}^{(x)} - \mathbf{h}_{i,w}^{(x)}||^2 \quad (15)$$

The final loss is calculated by combining this regularization loss with other losses, each with its the corresponding coefficients. We adopt the values for α and β from Kaneko and Bollegala (2021) with 0.2 and 0.8 respectively and set γ to 0.1.

$$L = \alpha L_i + \beta L_{reg}^{(debias)} + \gamma L_{reg}^{(prefix)} \quad (16)$$

We test the following parameters: number of tokens,

insert prefix in between CLS token and the input tokens.

during training for a downstream task, we train the model parameters and freeze the prefix

5 Experiments

5.1 Datasets

We construct $\mathcal{V}$ from a list of attributes and stereotypes [2] provided by Kaneko and Bollegala (2021). We also use the same News-commentary-v15 corpus [3] used to debias the model.

5.2 Evaluations

As an intrinsic evaluation metric, SEAT 6, 7, and 8 [4] are used. This set of tests all represent some form of gender bias (see Appendix C).

LPBS adjectives, occupations and LPBS based on stereotypes $\mathcal{V}$ [5].

The problem with LPBS in downstream tasks is that the masked language classifier cannot be used after training on a downstream task. We overcome this the following way, ..

To establish a baseline, we also construct a SEAT and LPBS test based on the attributes and stereotypes in $\mathcal{V}$ on which the model is trained[6]. We expect to see a high bias for these tests in the base model and low bias after the debiasing process. We refer to these tests as SEAT-$\mathcal{V}$ and LPBS-$\mathcal{V}$.

5.3 Hyper-parameters

DistilBERT, BERT, RoBERTa [7]

6 Discussion

7 Future work

- train with various sizes of prefixes, then the first token is the most important, and each extra token introduces more debiasing strength

- apply to LLM's

[2] https://github.com/kanekomasahiro/context-debias/tree/main/data

[3] https://data.statmt.org/news-commentary/v15/training-monolingual/

[4] https://github.com/W4ngatang/sent-bias/tree/master/tests

[5] https://github.com/m4urin/prefix-debiasing/blob/main/data/eval/seat/stereo.json

[6] https://github.com/m4urin/prefix-debiasing/tree/main/data/eval

[7] https://github.com/huggingface/transformers

Model	Type	SEAT-6	SEAT-7	SEAT-8	SEAT- $\mathcal{V}$
DistilBERT	base	†0.50	0.12	†0.51	†0.41
	finetune	-0.10	0.01	0.18	-0.02
	prefix	†0.35	0.27	-0.22	-0.13
BERT	base	†0.72	0.06	0.19	†1.17
	finetune	-0.01	-0.07	-0.11	0.11
	prefix	0.16	0.11	0.22	0.11
RoBERTa	base	0.09	0.06	0.17	0.10
	finetune	0.22	0.05	0.24	0.19
	prefix	0.18	-0.12	-0.11	0.09

Table 2: SEAT scores (0 is desirable). Negative values indicate negative associations (not desirable). Values that are significant at $\alpha < 0.05$ are marked with †.

Model	Type	LPBS Adjectives	LPBS Occupations	LPBS $\mathcal{V}$
DistilBERT	base	**0.45 ± 0.38**	**0.63 ± 0.57**	**0.65 ± 0.51**
	finetune	0.62 ± 0.47	0.77 ± 0.67	0.96 ± 0.76
	prefix	1.07 ± 0.73	0.96 ± 0.78	0.91 ± 0.64
BERT	base	0.56 ± 0.49	0.95 ± 0.87	1.04 ± 0.78
	finetune	0.67 ± 0.53	0.81 ± 0.70	0.80 ± 0.69
	prefix	**0.43 ± 0.33**	**0.48 ± 0.37**	**0.44 ± 0.35**
RoBERTa	base	0.93 ± 0.76	1.07 ± 0.84	1.04 ± 0.83
	finetune	**0.89 ± 0.66**	**0.90 ± 0.77**	**0.66 ± 0.54**
	prefix	1.48 ± 1.11	1.55 ± 1.16	2.00 ± 1.36

Table 3: LPBS scores (low score is desirable). Standard deviation is indicated with ± (low std is desirable).

Model	Type	Gender Accuracy (%)	Stereotype Accuracy (%)	Stereotype Confidence	p-value
DistilBERT	base	99.86	70.75	0.4115	0.0000
	finetune	99.18	60.4	0.3967	0.0000
	prefix	99.73	75.15	0.4251	0.0000
BERT	base	99.86	65.87	0.4432	0.0000
	finetune	81.08	56.79	0.2538	0.0000
	prefix	96.13	80.18	0.4125	0.0000
RoBERTa	base	99.76	77.69	0.4151	0.0000
	finetune	88.11	64.94	0.2787	0.0000
	prefix	53.6	53.71	0.0464	0.9546

Table 4: Gender probe

Model	Type	SST2	MRPC	RTE	WSC
DistilBERT	base	**85.09**	**70.59**	53.28	60.58
	finetune	51.61	68.38	47.6	53.85
	prefix	76.15	70.1	**55.02**	**64.42**
BERT	base	**88.53**	**72.3**	**58.08**	54.81
	finetune	50.46	68.38	46.29	**64.42**
	prefix	48.97	68.38	48.91	62.5
RoBERTa	base	**91.51**	**82.84**	**68.56**	55.77
	finetune	56.08	68.38	47.16	**63.46**
	prefix	47.48	68.38	47.16	**63.46**

Table 5: Glue performance on the test set.

8 Conclusion

Acknowledgements

Hi Ewoe, you made it to the end!

References

Christine Basta, Marta R Costa-Jussà, and Noe Casas. 2019. Evaluating the underlying gender bias in contextualized word embeddings. *arXiv preprint arXiv:1904.08783*.

Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. 2020. Language (technology) is power: A critical survey of" bias" in nlp. *arXiv preprint arXiv:2005.14050*.

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *Advances in neural information processing systems*, 29.

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

Yang Trista Cao, Yada Pruksachatkun, Kai-Wei Chang, Rahul Gupta, Varun Kumar, Jwala Dhamala, and Aram Galstyan. 2022. On the intrinsic and extrinsic fairness evaluation metrics for contextualized language representations. *arXiv preprint arXiv:2203.13928*.

Pengyu Cheng, Weituo Hao, Siyang Yuan, Shijing Si, and Lawrence Carin. 2021. Fairfil: Contrastive neural debiasing method for pretrained text encoders. *arXiv preprint arXiv:2103.06413*.

Kate Crawford. 2017. The trouble with bias. *NIPS 2017 Keynote*.

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

Pieter Delobelle, Ewoenam Kwaku Tokpo, Toon Calders, and Bettina Berendt. 2021. Measuring fairness with biased rulers: A survey on quantifying biases in pretrained language models. *arXiv preprint arXiv:2112.07447*.

Sunipa Dev, Tao Li, Jeff M Phillips, and Vivek Srikumar. 2020. Oscar: Orthogonal subspace correction and rectification of biases in word embeddings. *arXiv preprint arXiv:2007.00049*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Yuhao Du and Kenneth Joseph. 2020. Mdr cluster-debias: A nonlinear word embedding debiasing pipeline. In *International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation*, pages 45–54. Springer.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings. *arXiv preprint arXiv:1909.00512*.

Hila Gonen and Yoav Goldberg. 2019. Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them. *arXiv preprint arXiv:1903.03862*.

Wei Guo and Aylin Caliskan. 2021. Detecting emergent intersectional biases: Contextualized word embeddings contain a distribution of human-like biases. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 122–133.

Yue Guo, Yi Yang, and Ahmed Abbasi. 2022. Autodebias: 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)*, pages 1012–1023.

Souleiman Hasan and Edward Curry. 2017. Word re-embedding via manifold dimensionality retention. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 321–326.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.

Masahiro Kaneko and Danushka Bollegala. 2019. Gender-preserving debiasing for pre-trained word embeddings. *arXiv preprint arXiv:1906.00742*.

Masahiro Kaneko and Danushka Bollegala. 2021. Debiasing pre-trained contextualised embeddings. *arXiv preprint arXiv:2101.09523*.

Divyansh Kaushik, Eduard Hovy, and Zachary C Lipton. 2019. Learning the difference that makes a difference with counterfactually-augmented data. *arXiv preprint arXiv:1909.12434*.

Keita Kurita, Nidhi Vyas, Ayush Pareek, Alan W Black, and Yulia Tsvetkov. 2019. Measuring bias in contextualized word representations. *arXiv preprint arXiv:1906.07337*.

Anne Lauscher, Tobias Lüken, and Goran Glavaš. 2021. Sustainable modular debiasing of language models. *arXiv preprint arXiv:2109.03646*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Paul Pu Liang, Irene Mengze Li, Emily Zheng, Yao Chong Lim, Ruslan Salakhutdinov, and Louis-Philippe Morency. 2020. Towards debiasing sentence representations. *arXiv preprint arXiv:2007.08100*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021a. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021b. Gpt understands, too. *arXiv preprint arXiv:2103.10385*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Kaiji Lu, Piotr Mardziel, Fangjing Wu, Preetam Amancharla, and Anupam Datta. 2020. Gender bias in neural natural language processing. In *Logic, Language, and Security*, pages 189–202. Springer.

Thomas Manzini, Yao Chong Lim, Yulia Tsvetkov, and Alan W Black. 2019. Black is to criminal as caucasian is to police: Detecting and removing multiclass bias in word embeddings. *arXiv preprint arXiv:1904.04047*.

Rowan Hall Maudslay, Hila Gonen, Ryan Cotterell, and Simone Teufel. 2019. It's all in the name: Mitigating gender bias with name-based counterfactual data substitution. *arXiv preprint arXiv:1909.00871*.

Chandler May, Alex Wang, Shikha Bordia, Samuel R Bowman, and Rachel Rudinger. 2019. On measuring social biases in sentence encoders. *arXiv preprint arXiv:1903.10561*.

Nicholas Meade, Elinor Poole-Dayan, and Siva Reddy. 2021. An empirical survey of the effectiveness of debiasing techniques for pre-trained language models. *arXiv preprint arXiv:2110.08527*.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2021. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6):1–35.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Moin Nadeem, Anna Bethke, and Siva Reddy. 2020. Stereoset: Measuring stereotypical bias in pretrained language models. *arXiv preprint arXiv:2004.09456*.

Malvina Nissim, Rik van Noord, and Rob van der Goot. 2020. Fair is better than sensational: Man is to doctor as woman is to doctor. *Computational Linguistics*, 46(2):487–497.

Hadas Orgad and Yonatan Belinkov. 2022. Choose your lenses: Flaws in gender bias evaluation. *arXiv preprint arXiv:2210.11471*.

Ji Ho Park, Jamin Shin, and Pascale Fung. 2018. Reducing gender bias in abusive language detection. *arXiv preprint arXiv:1808.07231*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. 2020. Null it out: Guarding protected attributes by iterative nullspace projection. *arXiv preprint arXiv:2004.07667*.

Shauli Ravfogel, Michael Twiton, Yoav Goldberg, and Ryan D Cotterell. 2022. Linear adversarial concept erasure. In *International Conference on Machine Learning*, pages 18400–18421. PMLR.

Rachel Rudinger, Jason Naradowsky, Brian Leonard, and Benjamin Van Durme. 2018. Gender bias in coreference resolution. *arXiv preprint arXiv:1804.09301*.

Timo Schick, Sahana Udupa, and Hinrich Schütze. 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.

João Sedoc and Lyle Ungar. 2019. The role of protected class word lists in bias identification of contextualized word representations. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 55–61.

Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*.

Yarden Tal, Inbal Magar, and Roy Schwartz. 2022. Fewer errors, but more stereotypes? the effect of model size on gender bias. *arXiv preprint arXiv:2206.09860*.

- Applying [neutralization / projection to a bias subspace].
- A [static / non-contextualized] word embedding.

B Hyper-parameters

Parameter	Value	Description
prefix_mode	linear	linear transformation of the embedding in each layer
prefix_layers	all	linear transformation applied to each layer
prefix_n_tokens	8	number of prefix tokens
debias_method	kaneko	debiasing method / loss function
emb_pool	mean	pooling of multi-token words
epochs	3	number of epochs
batch_size	32	16 attribute sentences, 16 stereotype sentences
optimizer	AdamW	modifies implementation of weight decay in Adam
lr	5e-4	learning rate
lr_scheduler	linear	decreasing the learning rate linearly to 0 during training
warmup_steps	100	increase learning rate from 0 to lr in 100 steps

Table 6: Hyper-parameters descriptions.

Parameter	Debiasing		Probing		GLUE benchmark			
	Fine-tuning	Prefix-tuning	Intrinsic probe	MLM head	SST2	MRPC	RTE	WSC
debias_method	kaneko	kaneko	kaneko	kaneko	kaneko	kaneko	kaneko	kaneko
emb_pool	mean	mean	mean					mean
optimizer	AdamW	AdamW	AdamW	AdamW	AdamW	AdamW	AdamW	AdamW
epochs	3	3	3	3	1	4	4	20
batch_size	32	32	32	32	32	24	16	24
lr	2e-5	5e-4	1e-5	1e-5	2e-6	2e-6	5e-6	2e-6
lr_scheduler	linear	linear	linear	linear	linear	linear	linear	linear
warmup_steps	100	100	40	40	40	40	40	40
train_head			15%	15%	15%	15%	15%	15%
prefix_mode		linear						
prefix_layers		all						
prefix_n_tokens		8						

Table 7: Hyper-parameters used in the experiments. See Table 6 for the description of each parameter.

C Evaluations

SEAT	Target 1	Target 2	Attribute 1	Attribute 2	Size
6	Male names	Female names	Career	Family	64
7	Math	Arts	Male terms	Female terms	72
8	Science	Arts	Male terms	Female terms	80
$\mathcal{V}$	Male stereotypes	Female stereotypes	Male attributes	Female attributes	140

Table 8: Target and attribute categories for each SEAT.