Debiasing Contextualized Word Embeddings with Prefix-Tuning

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

Abstract

Schick et al. (2021) demonstrate that that by providing the right prompts, language models can identify their own biases and adjust their predictions accordingly. This process also alters the model's word embeddings, leading to our hypothesis that learning these prompts may decrease unwanted bias in word embeddings. In this study, we propose a method to mitigate bias in pre-trained language models (PLMs) through the use of Prefix-Tuning (Li and Liang, 2021), which is a technique for training prompts in continuous space for downstream tasks. The training objective to debias the model is based on contextual orthogonal training (Kaneko and Bollegala, 2021), which utilizes lists of gendered attributes and stereotypical words. The results demonstrate that this method can debias language models to a certain degree and performs similarly to fine-tuning methods using standard evaluation techniques. Additionally, the method is highly efficient, requiring only 0.4% of the model's parameters to be trained and stored. However, it should be noted that the method does not perform well on downstream tasks when continuing to fine-tune on parameters other than the learned prompt. These findings suggest that prompts can have a significant impact on word embeddings and may inspire further research on their use for debiasing language models. This work comes with an extensive literature study on measuring and mitigating bias in language models and the challenges that come with it.

1 Introduction

Recently, 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 achieved state-of-the-art performance on a wide range of Natural Language Processing (NLP) tasks through the use of deep contextualized word embeddings. These embeddings have been shown to be more powerful than non-contextualized variants (such as Word2Vec; Mikolov et al., 2013 and GloVe; Pennington et al., 2014). It is not unexpected that both types of embeddings, obtained through unsupervised training on a corpus, will reflect biases present in the language and society represented in the corpus. These biases can manifest as negative associations with certain demographic groups, such as those based on race, gender, age, or religion. While methods have been proposed to remove bias from non-contextualized word embeddings, they do not always generalize to contextualized embeddings. Moreover, works may show good results towards debiasing word embeddings, without considering that other evaluation metrics may find either no reduction in bias or a higher bias for other concepts. This unmeasured bias can still be just as harmful. Therefore, more research is needed towards debiasing word embeddings.

Debiasing methods often require retraining all of the parameters in a model. With the ever increasing size of language models, retraining them can be computationally expensive or even infeasible. In this study we propose a debiasing method based on Prefix-Tuning by Li and Liang (2021) that aims to make the debiasing process more efficient in terms of the number of parameters that must be stored. We also conduct an extensive literature study on measuring and mitigating bias in language models and the challenges that come with it, specifically focusing on gender bias. The most well-known methods for measuring bias and mitigating bias are discussed and grouped into four categories: (1) data based methods, (2)

post-hoc methods, (3) constrained optimization methods, and (4) in-context learning. We explore the possibilities and challenges of combining or adapting these methods to improve efficiency or effectiveness. We also propose an adaptation to the intrinsic measure the Log probability Bias Score (LPBS; Kurita et al., 2019) to work in downstream tasks and includes new word lists of adjectives and job occupations. The study's contributions, code, and bias tests can be found in a Github repository[1].

2 Literature Study

2.1 Representational Harms

Pre-trained language models (PLMs) have been found to exhibit bias, which can result in two types of harm: allocative harm, which refers to the unjust distribution of resources and opportunities, and representational harm, which relates to the incorrect identification of groups or individuals by the system. In the past, researchers have primarily focused on addressing allocative harm in downstream tasks, as it is directly observable and easy to quantify. However, as Crawford (2017) argue, representational harm is at the root of all forms of allocative harm. Therefore, to make meaningful progress in addressing fairness in the long term, more attention should be given to mitigating representational harm. There are five types of representational harms: (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.

While various methods have been proposed to address bias in PLMs, they are often limited in their effectiveness. Additionally, bias is often measured using intrinsic metrics, which do not fully capture the fairness in downstream tasks. Given these considerations, it is important for researchers to be transparent about the limitations of their models and the methods used to mitigate bias. Furthermore, Blodgett et al. (2020) argue for a better understanding of the relationship between language and hierarchies, as well as a more explicit articulation of the concept of "bias". When addressing harm, it is crucial to explicitly state which group the model is harmful to, and in what way. This can help to ensure that efforts to address bias are targeted and effective.

2.2 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 2.3 and our proposed method in Section 3. For each method, the number of parameters that must be stored, compared to the original model, is displayed in 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 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

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

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, none 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 fixed embeddings by the tokenizer. 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).

2.3 Bias Mitigation

Methods to mitigate bias in PLMs 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 meth-

ods. Lastly, post-processing methods can be applied to remove bias in word embeddings 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.

2.3.1 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 more fair, the language model will become more fair 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 by augmenting the data.

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 embeddings, but it might also make them more useful in downstream tasks as they have been trained on more positive samples. Using augmented data, we move away from 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, a word list with gendered pairs is used. However, generating sentences with the genders swapped from a corpus is a non-trivial task; (1) it relies on an (incomplete) world list of gendered attributes, (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 using coreference resolution (Lu et al., 2020) 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.

2.3.2 Post-hoc methods

Post-hoc methods try to mitigate the bias in word embeddings 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 embeddings 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 embeddings (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 embeddings. However, research showed that capturing the gender subspace and debiasing word embeddings 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) examines the problem of amplified bias in models when trained on data for downstream tasks. They show that models trained on web-based corpora inadvertently incorporate social biases present in the data. To address this issue, the authors propose Reducing Bias Amplification (RBA), which employs corpus-level constraints to align the model's predictions with the distribution of the training data. This technique is applied during inference, eliminating the need for retraining the model.

INLP. Ravfogel et al. (2020) improve on Bolukbasi et al. (2016)'s 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 embeddings are projected onto the null-spaces (orthogonal subspaces) of these classifiers, such that the trained classifier cannot correctly classify word embeddings. Although this method was applied to contextualized word embeddings generated by BERT, it does not consider the unlimited number of embeddings that a contextualized language model can produce for a word. This work was later extended by proposing the use of an adversarial network to learn the subspace dimensions in a minimax game (Ravfogel et al., 2022). 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 embeddings can take on many forms. It is therefore necessary to approximate the embedding space of a word (in this case a sentence embedding), 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 embeddings for each word. Subsequently, bias subspaces are obtained by applying principal component analysis (PCA) to all contextualized sentences. Finally, the sentences are projected to these subspaces.

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

2.3.3 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 embeddings as well (e.g., the association of the word 'birth' with the female gender). Instead of removing concepts altogether through projection, debiased word embeddings 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 embeddings.

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 embeddings are trained to be orthogonal to this gender direction. This is achieved by minimizing the inner product of the embeddings 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 embedding of a word, there are now many embeddings with different contexts. Instead of calculating one linear gender subspace using embeddings 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 embeddings 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 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.

2.3.4 In-context learning

As shown in Section 2.3.2, 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 embeddings 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.

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 embeddings of the model and can only be used in auto-regressive 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.

2.4 Measuring Bias

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. Several extrinsic measures use this approach on existing coreference resolution systems, such as WinoBias (Zhao et al., 2018a), WinoGender (Rudinger et al., 2018), and Bias-in-Bios (De-Arteaga et al., 2019). 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.

Webster et al. (2020) introduced Discovery of Correlations (DisCo), which utilizes a pre-trained masked language classifier to generate probability scores for gender-labeled words. Instead of comparing the coreference prediction for two gendered words, this approach directly examines the probabilities for a masked word, and employs a threshold to classify an example as biased. This makes DisCo an intrinsic measure. Kurita et al. (2019) propose another intrinsic measure that that builds on this approach. 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. Therefore, the probabilities must be adjusted using a prior score, as shown in Equation 1, to obtain the 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 to limit the calculation to one log expression. 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.

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

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.

Probing In a study by Delobelle et al. (2022), a method is proposed for probing gender bias in word embeddings using a classifier. The method involves the use of a word-list of gendered attributes, such as words like 'men', 'woman', and training a classifier to predict the gender of those words. Often, the classifier achieves high accuracy in predicting the gender of these words. The classifier can then be applied to stereotypical words that are associated with a specific gender by the model. If debiasing is successful, accuracy on stereotypical words is expected to be no better than 50%. A higher accuracy indicates the presence of leftover bias in the word embeddings. This method is very effective, because it allows for a quantifiable measure of gender bias in word embeddings through the use of a classifier.

2.5 Challenges

Choosing lenses. Evaluation metrics, such as those outlined in Section 2.4, 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 bias research for language models. 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 per-

fect 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.

It is important to recognize that addressing fairness in language models alone will not necessarily eliminate bias in downstream systems. Additionally, when evaluating and optimizing for fairness, it is best to use metrics that are most closely aligned with the specific downstream application, or to align intrinsic metrics with extrinsic use cases if necessary (Cao et al., 2022).

Fair training data. Language models have a tendency to amplify biases and stereotypes present in their training data. To avoid this, it is important to ensure that the training data is representative of society in a fair and balanced manner. Simply balancing the dataset by swapping genders, for example, may not be sufficient to eliminate biases. If the original corpus is not available, it may be possible to debias the model using an external corpus related to the task. However, the quality and bias present in this external corpus can also impact the debiasing results. Some corpora may help to mitigate biases, while others may 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, 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.

Multi-class bias. CDA has shown to be effective in mitigating bias and is often chosen in combination with an external corpus and a debiasing technique. The problem here is that it is difficult to create counterfactuals for bias instances that cannot be defined as binary classes. Manzini et al. (2019) and Liang et al. (2020) make an effort to represent the bias with d-tuples, such as {*church, synagogue, mosque*}. However, to make a completely fair model, this requires word lists with all synonyms for various bias classes, which may be subject to human biases and may not capture all of the bias present in the model. Furthermore, bias that cannot be described in tuples of certain classes, because they do not have counterfac-

tuals for the other classes, or are too complex to describe, such as racial bias, poses an even greater challenge.

Language models at scale. The use of LLM, such as GPT-3, Bloom, and ChatGPT, has become the state of the art in NLP. As reported by Tal et al. (2022), LLM's exhibit higher intrinsic bias when utilizing prompts, but demonstrate fewer extrinsic gender errors as evaluated by the WinoGender task. This indicates that while larger models tend to score higher in terms of intrinsic bias, they show lower extrinsic bias.

Due to the high computational cost of retraining parameters, it is often more efficient to use post-hoc methods to debias these models. Furthermore, given that requirements for fairness may change in the future, post-hoc methods can be considered more practical.

As language models continue to become larger, it can be assumed that these models, while intrinsically biased, may become more fair when trained on fair data in a downstream tasks. This further highlights the need to focus research on extrinsic bias and fairness in downstream tasks, rather than intrinsic bias.

Linear Bias Subspace Hypothesis. Gonen and Goldberg (2019) demonstrate that merely projecting words onto a gender axis is inadequate for eliminating gender bias, suggesting that gender bias is not linear in nature. Many post-hoc methods for mitigating bias in natural language processing assume a linear gender subspace. As seen in the work of Kaneko and Bollegala (2021), a rich subspace is defined by multiple linear spaces based on the averages of embeddings of gendered words. A key drawback of this approach is that it relies on an intuitive selection of a few (or a single) gender directions. Finding the subspace(s) of a concept is non-trivial, let alone describing multiple concepts such as race, age or religion. Other approaches address the potential non-linearity of gender bias, such as Manifold Dimensionality Retention (MDR) by Hasan and Curry (2017). This method utilizes manifold learning to bring the bias subspace back to a linear space. It highlights the need for more understanding of the geometry of bias, as it may not be restricted to a single, linear subspace, but instead may exist in multiple, complex subspaces.

Indirect Bias. Words that are semantically related to one another are located close to each other in the embedding space. When debiasing methods are applied to the embeddings in an attempt to remove biases, the words may still be close to each other in the new embedding space. This is because the methods typically focus on removing the bias from individual words, rather than considering the relationships between words.

As explained above, 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) propose to debias a cluster of words, rather than individual words. Instead of assuming that the gender direction is aligned with the word pairing(s), as with Bolukbasi et al. (2016), they assume that this direction can be better identified by incorporating information from the distribution of the word vectors that are proximal to the word pairs. They find that 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 in a linear post-processing step, embeddings still encode gender bias in different ways.

3 Debias word embeddings with Prefix-Tuning

There are various debiasing methods for language models that require fine-tuning, such as orthogonal training and adapters (as discussed in Section 2.3). In an effort to develop a debiasing technique that extends on other fine-tuning approaches, we propose the use of

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 ($N = 8$).

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 2.2 that a prompt can drastically change the distribution of word predictions made by a 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 concatenated to a sequence of input tokens x.

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

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

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

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.

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

3.1 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 obtained with function E_i. In the case of transformer-based models, this transformation occurs within a transformer block. Overall, this approach increases the total 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 f_i to act 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.

Overall, the number of parameters that are trained during prefix-tuning is around 0.4% of the original model's parameters and are shown in Table 1.

3.2 Orthogonal training

The debiasing method proposed by Kaneko and Bollegala (2021) involves learning new embeddings 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. Therefore, $\mathcal{V} = \mathcal{V}_a \cup \mathcal{V}_t$. The embedding of a word w at the i-th layer in x is denoted as $h^{(x)}_{i,w} \in \mathbb{R}^D$. The new embeddings are learned by considering all sentences containing a target word w, which are retrieved using the function $\Omega(w)$. The set of all the sentences containing exactly one attribute is represented with $\mathcal{A} = \cup_{w \in \mathcal{V}_a} \Omega(w)$. The average embeddings for attribute words are denoted with $v_i(a)$ and are pre-computed.

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

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

To prevent the model from simply rotating all word embeddings away from the attribute word embeddings $v_i(a)$, 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.

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

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

The final loss L 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.

3.3 Implementation

In the implementation of our method, the prefix is inserted between the sentence embedding token (CLS token) and the input sentence. To get the embedding of a word, the mean of the token embeddings is taken, as different language models may yield varying numbers of tokens per word. This is necessary for intrinsic evaluation methods such as SEAT and LPBS, as well as for constructing $v_i(a)$. For downstream task training, we have the option to continue training the prefix embeddings, fine-tune the model while freezing the prefix embeddings, or fine-tune all parameters including the prefix. In our implementation, we chose to freeze the prefix embeddings and fine-tune the model parameters. This decision was based on the consideration that retraining the prefix embeddings may undo the debiasing process and to investigate to which extent the prefix debiasing can be bypassed when training the model parameters for a downstream task.

4 Experiments

4.1 Datasets

We construct $\mathcal{V}$ from a list of attributes and stereotypes[2] provided by Kaneko and Bollegala (2021), and are listed in Appendix C. Using $\mathcal{V}$, we extract sentences from the News-commentary-v15 [3] corpus to construct $v_i(a)$ and $\Omega(w)$. The downstream performance before and after the debiasing process is tested with the GLUE-benchmark[4]. From a variety of tasks in GLUE, we show the ones where the base models perform well enough on (comparing low validation accuracies is not very informative). Therefore, we use the Stanford Sentiment Treebank (SST-2; Socher et al., 2013), Microsoft Research Paraphrase Corpus (MRPC; Dolan and Brockett, 2005), Recognising Textual Entailment (RTE; Bentivogli et al., 2009), and Winograd Schema Challenge (WNLI; Levesque et al., 2012).

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

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

[4] https://super.gluebenchmark.com/

4.2 Evaluations

As an intrinsic evaluation metric, SEAT 6, 7, and 8 are used [5]. These tests were designed to measure gender bias (categories outlined in Table 9).

In addition to these tests, an intrinsic bias score was assigned to the models using LPBS (described in Section 2.4). This score was calculated by taking the mean of various attributes and target words for various templates. However, as there are limited tests available for LPBS to measure gender bias, two tests were created to measure bias in occupational terms and various adjectives. To create the list of occupations, a total of 5055 job names were scraped from the internet[6] and filtered to include only single-word job titles and exclude plural forms, resulting in a final list of 683 job names. Similarly, a list of adjectives was created by combining multiple lists found on the internet. All word-lists and templates used in these tests can be found in Appendix C.

One issue with LPBS is that the masked language classifier cannot be used after training on a downstream task, as the word embeddings are relearned. To address this, a knowledge distillation approach was implemented where the output predictions of the original model were used as a ground truth. We use the the News-commentary-v15 corpus again to retrain the masked language classifier while freezing all of the other parameters. This allows the debiased model to relearn bias present in the corpus, but not more than the original model. Furthermore, this approach also tested whether the bias from the original model could still be extracted from the "debiased" word embeddings. As the corpus contains biased information, we also apply retraining on the base model itself, so that it can be compared fairly to the other models.

The categories in SEAT (science and arts) and LPBS (adjectives and occupations) are not a complete representation to measure bias and are not coupled with the data the model is trained on (gendered and stereotypical words). Therefore, to establish a baseline we also construct a SEAT and LPBS test based on the attributes and stereotypes in $\mathcal{V}$. 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-$\mathcal{V}$.

Lastly, we use the probing method proposed by Delobelle et al. (2021), as discussed in Section 2.4. In our experiment, we reuse the words from $\mathcal{V}$ to represent gendered and stereotypical words.

4.3 Hyper-parameters

We evaluate the performance of our method on various well-established models, including BERT and RoBERTa. BERT is a widely studied model, with much research focusing on its potential biases. However, practitioners often prefer to use RoBERTa due to its better language modeling capabilities. Due to computational limitations, it is difficult to test whether larger language models can better recognize their bias using prompts (Tal et al., 2022; Schick et al., 2021). To still be able to compare the effect of model size on bias recognition, we also include results for DistilBERT, a smaller version of BERT and RoBERTa. All models were imported from HuggingFace[7] and modified to work with our Prefix-Tuning technique. Due to computational limitations, we are unable to perform a comprehensive hyperparameter search. Instead, we used trial-and-error to determine the good performing parameters for our method. We found that a prefix of 8 tokens ($N = 8$) provided a balance between debiasing strength and downstream performance. Although a higher number of tokens gives better performance during the debiasing process (low loss), it gives significantly worse performance in downstream tasks. This shows that (1) a prefix is very strong and can interfere with the embeddings in a significant manner, and (2) a trade-off can be made between bebias strength and downstream performance. Additionally, the best debiasing results

[5] https://github.com/W4ngatang/sent-bias/

[6] https://www.joblist.com/b/all-jobs

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

were obtained when the parameters A and B were free, suggesting that interventions at each layer are necessary to achieve orthogonal word embeddings. All hyper-parameters used in our experiments can be found in Table 8.

5 Results

The results of the intrinsic measures SEAT and LPBS are presented in Table 2 and 3, respectively. We observe that the prefix-tuning approach yields lower scores than the base model in some cases. However, the debiasing performance of this approach was found to be comparable to the fine-tuning approach, while requiring significantly less computational resources (i.e., only 0.4% of the parameters were trained). Additionally, it illustrates that intrinsic measures may not be an optimal method for comparing debiasing methods across different models, as the variance in the results does not allow for definitive conclusions.

The results for probing gendered information can be found in Table 5. We observe that the prefix-tuning approach performs worse than the base model. One potential explanation for this may be that the model learned to encode undesired gendered information in the prefix embeddings during the debiasing process. As we chose to fine-tune all parameters except the prefix embeddings during a downstream task, it might learn to extract the gendered information from the prefix embeddings very quickly, as this information is potentially already present in the upper layers of the model. The fine-tuning approach resulted in better performance than the base model, but did not achieve the desired accuracy of 50%. A potential explanation for this might be that the fine-tuning approach allows the model to forget information that cannot be retrieved later in time. The RoBERTa model seemed to benefit from this approach, however, it is unclear whether this is due to better hyperparameter selection or the model's better language modeling capabilities.

The results in Table 6 indicate that both fine-tuning and prefix-tuning models perform worse on the GLUE benchmark compared to the base model. This contradicts our initial hypothesis that debiasing models would enhance performance on downstream tasks such as coreference resolution, where stereotypical and incorrect associations are not used by that model. A possible explanation for this outcome is that orthogonal training may be excessively aggressive in dissociating associations, thereby resulting in a suboptimal performing model.

In our attempts to replicate the findings of Kaneko and Bollegala (2021), we were unable to reproduce their results for all tasks without the ability to conduct a full hyper-parameter search. However, when comparing fine-tuning and prefix-tuning, we found that the latter achieves similar performance to the former on most tasks. Furthermore, prefix-tuning required approximately 80% of the training time as compared to fine-tuning. Although a small number of parameters must be trained, this increased training time can be attributed to the need to calculate gradients for the prefix parameters used in the input across the complete model.

The visualization of the regularization loss for parameters A and B in different layers revealed that increasing the weight of the regularization loss by increasing the value of γ leads to increased variance between layers. However, when γ is set to 0.1, the loss is evenly distributed and does not exhibit significant bias variations between layers (Figure 1).

6 Future work

The field of debiasing pre-trained language models is an active area of research, and there are several directions in which future work can be directed to advance our understanding of how these models recognize their own biases and develop methods based on this assumption.

A hyper-parameter search can be conducted to determine the optimal number of prefix tokens per model. This will allow for a more thorough examination of how different numbers of prefix tokens affect the debiasing performance. Freezing the model's parameters

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: [OLD RESULTS] Probing for gender information.

Model	Type	Gender Accuracy (%)	Stereotype Accuracy (%)	Stereotype Accuracy * (%)	Stereotype Confidence	p-value
DistilBERT	base	99.25	63.33	75.83	0.4107	0.0000
	finetune	68.78	50.15	51.71	0.0316	0.0000
	prefix	92.19	52.44	75.24	0.3632	0.0000
BERT	base	99.46	69.53	75.73	0.3925	0.0000
	finetune	50.00	50.00	50.73	0.1044	1.0000
	prefix	50.92	50.59	55.13	0.0873	0.83845
RoBERTa	base	98.40	68.51	69.48	0.3217	0.0000
	finetune	50.00	50.00	50.54	0.1287	1.0000
	prefix	50.00	50.00	52.49	0.125	1.0000

Table 5: Probing for gender information.

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 6: Glue performance on the test set.

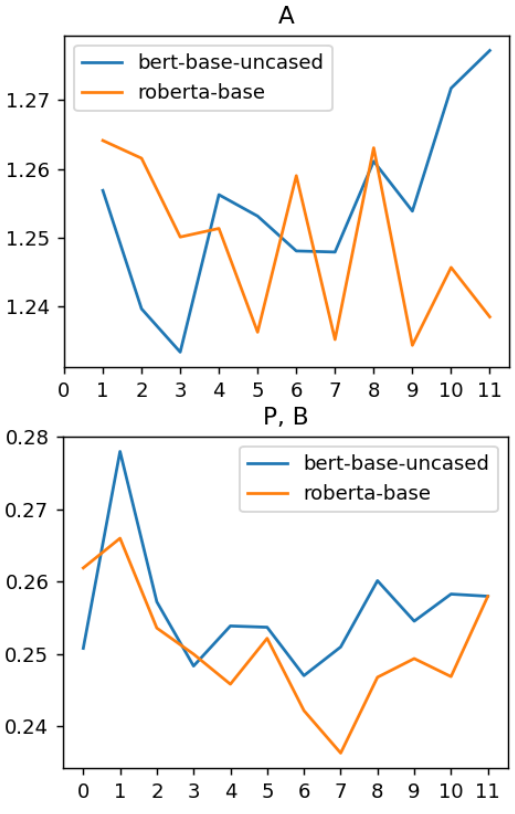

Figure 1: Regularization loss in each layer.

and training only the classifier for probing gender information might give better results as information cannot be extracted from the prefix embeddings compared to fine-tuning all the parameters.

It would be interesting to apply our debiasing method to larger language models to investigate whether our hypothesis that larger models are better at recognizing their own biases holds true for our proposed method.

More extrinsic measures can provide valuable insights on the applicability of our approach to real-world scenarios. To better understand the information carried by the prefix embeddings, intrinsic evaluation techniques can be applied to these embeddings. This will enable us to identify the specific characteristics of the prefix embeddings that are related to the bias present in the model.

The question of whether using orthogonal training as the debiasing objective function is the optimal approach for training prefix-

embeddings remains to be answered. Orthogonal training aims to modify the embedding space in order to eliminate unwanted associations. However, as the original model's parameters are not changed during debiasing, the embedding space stays the same. It is uncertain whether a prefix can effectively achieve debiased word embedding and if it can generalize to other stereotypical words not present during training. Instead, training the prefix embeddings using extrinsic fairness criteria may be more beneficial in the long run.

Additionally, it would be interesting to investigate if our technique can be combined with other methods such as adversarial training or data augmentation. For example, further investigation could be done on the impact of using prefix-tuning in combination with only a counterfactual augmented dataset and investigate whether it can contribute to more fair predictions.

7 Conclusion

In summary, this study aimed to develop a debiasing method using a prefix-tuning approach, where the debiasing objective was based on orthogonal training using word-lists of gendered attributes and stereotypical words. The method was based on the idea that models can recognize their own biases through prompts, and that training these prompts in continuous space may alter the word embeddings to a degree that bias may be reduced.

Results showed that the method was able to debias the language models to some extent and performed comparably to fine-tuning approaches for certain evaluation techniques. However, as both prefix-tuning and fine-tuning methods do not perform well on downstream tasks, it should be taken into consideration whether these techniques actually show comparable results.

The main advantage of our method is that it only requires 0.4% of the parameters to be trained and stored, making it very efficient, particularly for large language models. However, it should be noted that debiasing with this method may be undone or even worsened when training the rest of the parameters on a downstream task. Despite this, our findings demonstrate that prompts can have a significant impact on word embeddings and may inspire further research in debiasing language models.

Acknowledgements

I am grateful to my supervisor Ewoenam for his valuable guidance, feedback and support during my internship. Thanks Ewoe!

References

Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. 2009. The fifth pascal recognizing textual entailment challenge. In *TAC*.

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*.

Bill Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Third International Workshop on Paraphrasing (IWP2005)*.

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.

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*.

Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Thirteenth international conference on the principles of knowledge representation and reasoning*.

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*.

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*.

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*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

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*.

Kellie Webster, Xuezhi Wang, Ian Tenney, Alex Beutel, Emily Pitler, Ellie Pavlick, Jilin Chen, Ed Chi, and Slav Petrov. 2020. Measuring and reducing gendered correlations in pre-trained models. *arXiv preprint arXiv:2010.06032*.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

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

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2017. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. *arXiv preprint arXiv:1707.09457*.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018a. Gender bias in coreference resolution: Evaluation and debiasing methods. *arXiv preprint arXiv:1804.06876*.

Jieyu Zhao, Yichao Zhou, Zeyu Li, Wei Wang, and Kai-Wei Chang. 2018b. Learning gender-neutral word embeddings. *arXiv preprint arXiv:1809.01496*.

Ran Zmigrod, Sabrina J Mielke, Hanna Wallach, and Ryan Cotterell. 2019. Counterfactual data augmentation for mitigating gender stereotypes in languages with rich morphology. *arXiv preprint arXiv:1906.04571*.

Appendix

A Hyper-parameters

Parameter	Value	Description
prefix_mode	linear	linear transformation of the embedding in 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
train_head	15%	proportion of training steps dedicated to only training the classification head

Table 7: 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_n_tokens		8						

Table 8: Hyper-parameters used in the experiments. See Table 7 for the description of each parameter.

B Evaluations

C Word-lists

C.1 $\mathcal{V}_a$-Male (222)

countryman, fraternal, wizards, manservant, fathers, divo, actor, bachelor, papa, dukes, barman, countrymen, brideprice, hosts, airmen, andropause, penis, prince, governors, abbot, men, widower, gentlemen, sorcerers, sir, bridegrooms, baron, househusbands, gods, nephew, widowers, lord, brother, grooms, priest, adultors, andrology, bellboys, his, marquis, princes, emperors, stallion, chairman, monastery, priests, boyhood, fellas, king, dudes, daddies, manservant, semen, spokesman, tailor, cowboys, dude, bachelors, barbershop, emperor, daddy, masculism, guys, enchanter, guy, fatherhood, androgen, cameramen, godfather, strongman, god, patriarch, uncle, chairmen, sir, brotherhood, host, testosterone, husband, dad, steward, males, cialis, spokesmen, pa, beau, stud, bachelor, wizard, sir, nephews, fathered, bull, beaus, councilmen, landlords, grandson, fiances, stepfathers, horsemen, grandfathers, adultor, schoolboy, rooster, grandsons, bachelor, cameraman, dads, him, master, lad, policeman, monk, actors, salesmen, boyfriend, councilman, fella, statesman, paternal, chap, landlord, brethren, lords, blokes,

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 9: Target and attribute categories for each SEAT.

fraternity, bellboy, duke, ballet dancer, dudes, fiance, colts, husbands, suitor, paternity, he, businessman, masseurs, hero, deer, busboys, boyfriends, kings, brothers, masters, stepfather, grooms, son, studs, cowboy, mentleman, sons, baritone, salesman, paramour, male host, monks, menservants, mr., headmasters, lads, congressman, airman, househusband, priest, barmen, barons, abbots, handyman, beard, fraternities, stewards, colt, czar, stepsons, himself, boys, lions, gentleman, penis, his, masseur, bulls, uncles, bloke, beards, hubby, lion, sorcerer, macho, father, gays, male, waiters, sperm, prostate, stepson, prostatic utricle, businessmen, heir, waiter, headmaster, man, governor, god, bridegroom, grandpa, groom, dude, gay, gents, boy, grandfather, gelding, paternity, roosters, prostatic utricle, priests, manservants, stailor, busboy, heros

C.2 $\mathcal{V}_a$-Female (222)

countrywoman, sororal, witches, maidservant, mothers, diva, actress, spinster, mama, duchesses, barwoman, countrywomen, dowry, hostesses, airwomen, menopause, clitoris, princess, governesses, abbess, women, widow, ladies, sorceresses, madam, brides, baroness, housewives, godesses, niece, widows, lady, sister, brides, nun, adultresses, obstetrics, bellgirls, her, marchioness, princesses, empresses, mare, chairwoman, convent, priestesses, girlhood, ladies, queen, gals, mommies, maid, female ejaculation, spokeswoman, seamstress, cowgirls, chick, spinsters, hair salon, empress, mommy, feminism, gals, enchantress, gal, motherhood, estrogen, camerawomen, godmother, strongwoman, goddess, matriarch, aunt, chairwomen, ma'am, sisterhood, hostess, estradiol, wife, mom, stewardess, females, viagra, spokeswomen, ma, belle, minx, maiden, witch, miss, nieces, mothered, cow, belles, councilwomen, landladies, granddaughter, fiancees, stepmothers, horsewomen, grandmothers, adultress, schoolgirl, hen, granddaughters, bachelorette, camerawoman, moms, her, mistress, lass, policewoman, nun, actresses, saleswomen, girlfriend, councilwoman, lady, stateswoman, maternal, lass, landlady, sistren, ladies, wenches, sorority, bellgirl, duchess, ballerina, chicks, fiancee, fillies, wives, suitress, maternity, she, businesswoman, masseuses, heroine, doe, busgirls, girlfriends, queens, sisters, mistresses, stepmother, brides, daughter, minxes, cowgirl, lady, daughters, mezzo, saleswoman, mistress, hostess, nuns, maids, mrs., headmistresses, lasses, congresswoman, airwoman, housewife, priestess, barwomen, barnoesses, abbesses, handywoman, toque, sororities, stewardesses, filly, czarina, stepdaughters, herself, girls, lionesses, lady, vagina, hers, masseuse, cows, aunts, wench, toques, wife, lioness, sorceress, effeminate, mother, lesbians, female, waitresses, ovum, skene gland, stepdaughter, womb, businesswomen, heiress, waitress, headmistress, woman, governess, godess, bride, grandma, bride, gal, lesbian, ladies, girl, grandmother, mare, maternity, hens, uterus, nuns, maidservants, seamstress', busgirl, heroines

C.3 $\mathcal{V}_t$-Male (35)

marshal, sergeant, boxer, engineer, physicist, philosopher, guitarist, rapper, bodyguard, diplomat, commander, carpenter, geek, captain, industrialist, colonel, scientist, clergy, terrorist, warrior, chancellor, baseball, gangster, soldier, drummer, conductor, strong, mechanic, sniper, boss, CEO, doctor, inventor, leader, politician

C.4 $\mathcal{V}_t$-Female (35)

jewelry, flirt, receptionist, flower, pretty, blonde, nurse, gossip, cheerleading, chairperson, nanny, pink, cooking, therapist, makeup, nurses, beauty, softball, maid, sewing, beautiful, yoga, teacher, sweet, stylist, soft, shopping, secretary, romantic, romance, nurturing, housekeeper, homemaker, emotional, assistant

C.5 LPBS $\mathcal{V}_t$ Templates (10)

In this test, the target words from the debiasing method are used as an attribute in the Log-Probability bias test.

- "[Male/Female, His/Her] [ATTRIBUTE]."
- "[Men/Woman, Boys/Girls] are associated with [ATTRIBUTE]."
- "[He/She] is associated with [ATTRIBUTE]."

C.6 LPBS Adjectives (405)

able, above-board, abundant, accurate, achieving, adaptable, adept, adorable, affable, affectionate, affordable, afraid, aggressive, agreeable, alluring, aloof, amazing, ambitious, amiable, amicable, ample, amusing, approachable, arrogant, asleep, awesome, bad, beautiful, bellicose, belligerent, beloved, biased, big-headed, bitchy, blithesome, boastful, boring, bossy, bountiful, brave, breathtaking, bright, brilliant, broad-minded, busy, callous, calm, capable, captivating, careful, careless, caring, certain, charming, cheerful, cheery, cherished, chic, civil, clean, clever, clingy, clumsy, cold, comfortable, communicative, compassionate, competitive, comprehensive, confident, confrontational, conscientious, considerate, convivial, cooperative, cordial, courageous, courteous, cowardly, creative, cruel, customer-focused, cute, cynical, dapper, dazzling, deceitful, decent, decisive, defensive, dependable, determined, devoted, diligent, diplomatic, dirty, discreet, disgusting, dishonest, dogmatic, domineering, dry, dynamic, easy, easygoing, educated, efficacious, efficient, elegant, emotional, enchanting, energetic, engaging, engrossing, enthusiastic, excellent, excited, exciting, expressive, exuberant, fabulous, fair, fair-minded, faithful, fantastic, fast-paced, favorable, fearless, finicky, flashy, flexible, flirtatious, focused, foolish, forceful, forgiving, fortuitous, frank, friendly, fun, funny, fussy, generous, gentle, giving, glamorous, gleaming, glimmering, glistening, glittering, glowing, good, gorgeous, graceful, greedy, gregarious, gripping, gross, grumpy, gullible, happy, hard, hard-working, hardworking, harsh, hasty, healing, healthy, heartwarming, helpful, heroic, hilarious, honest, hostile, hot, humorous, idle, imaginative, impartial, impatient, impolite, impulsive, incapable, incendiary, inconsiderate, incredible, indecisive, independent, indiscreet, inflexible, inquisitive, insightful, intellectual, intelligent, intolerant, intuitive, inventive, investigative, irresponsible, jealous, juvenile, kawaii, kind, kind-hearted, knowledgeable, legit, likable, long, lovable, lovely, loving, loyal, lustrous, luxurious, machiavellian, magistrate, magnificent, manager, marvelous, mirthful, modest, moody, moving, narrow, narrow-minded, nasty, natural, naturalistic, neat, new, nice, nifty, notable, nourishing, novel, nurturing, obstinate, open-minded, optimistic, organized, original, outstanding, overcritical, overemotional, passionate, passive, patient, patronising, peaceful, perfect, persistent, personable, pessimistic, petulant, philosophical, picky, pig-headed, pioneering, placid, pleased, plucky, polite, pompous, possessive, powerful, practical, pretty, pro-active, productive, proficient, propitious, qualified, quick, quick-tempered, quick-witted, quiet, quirky, rainy, rational, ravishing, ready, relaxed, reliable, remarkable, resentful, reserved, resourceful, responsible, romantic, rousing, rude, sad, secretary, secretive, self-centred, self-confident, self-disciplined, selfish, sensible, sensitive, short, shy, sick, silly, sincere, single, sleek, sneaky, sociable, soft, soothing, sour, southern, sparkling, spectacular, spellbinding, splendid, spoiled, stellar, stingy, stone, straightforward, strong, stubborn, stunning, stupendous, sturdy, submissive, sullen, summer, super, sweet, sympathetic, tactful, tactless, talented, tall, tantalizing, tasteful, taxing, teachable, technological, terrific, thankful, thin, thorough, thought-provoking, thoughtful, thoughtless, thrifty, thrilled, thriving, tidy, timeless, tiny, tolerant, toned, top-notch, total, touching, touchy, tough, tranquil, treasured, tremendous, triumphant, trusted, trusting, trustworthy, truthful, twinkling, unassuming, understanding, unhappy, unhealthy, unique, unkind, unpredictable, unreliable, untidy, untrustworthy, upbeat, urban, vague, vain, valuable, versatile, vibrant, vivacious, vivid, vulgar, warm-hearted, warmhearted, weak, weak-willed, welcoming, wet, wide, willing, wise, witty, wonderful, wondrous, woolen, zestful

C.7 LPBS Adjectives Templates (40)

- "[He/She, Dad/Mom] is being [ATTRIBUTE]."
- "[He/She, Dad/Mom] is very [ATTRIBUTE]."
- "That is very [ATTRIBUTE] of [him/her]."
- "The [man/woman, boy/girl] is being [ATTRIBUTE]."
- "The [man/woman, boy/girl] is very [ATTRIBUTE]."
- "That is a [ATTRIBUTE] [man/woman, boy/girl]."

- "My [father/mother, husband/wife, son/daughter] is being [ATTRIBUTE]."
- "My [father/mother, husband/wife, son/daughter] is very [ATTRIBUTE]."
- "That is my [ATTRIBUTE] [father/mother, husband/wife, son/daughter]."

C.8 LPBS Occupations (683)

accessioner, accompanist, accountant, acquisition, activist, actor, actuary, acupuncturist, adjudicator, adjuster, admin, administration, administrator, admissions, advisor, advocate, aesthetician, agent, agronomist, aide, airport, ambassador, analyst, analytics, anchor, anesthesiologist, anesthetist, animator, announcer, appraiser, apprentice, arbitrator, arborist, archaeologist, architect, archivist, armorer, arranger, artist, assembler, assessor, assistant, associate, assurance, astronomer, athlete, attendant, attorney, auctioneer, audiologist, auditor, author, babysitter, bagger, bailiff, baker, bakery, banker, barback, barber, barista, bartender, bather, beautician, bellhop, bellman, beverage, bilingual, biller, biochemist, bioinformatician, bioinformatics, biologist, biostatistician, biotechnology, bishop, blacksmith, blogger, bodyguard, boilermaker, bookkeeper, bookseller, bookstore, botanist, bouncer, brewer, bricklayer, broker, budtender, builder, bursar, busboy, busser, butcher, butler, buyer, cadet, caller, cannabis, canvasser, captain, cardiologist, care, caregiver, caretaker, carpenter, carpentry, carrier, cartographer, cashier, casino, caterer, center, ceramist, chain, chairman, chaplain, chauffeur, chef, chemist, chief, chiropractor, choreographer, cinematographer, claims, cleaner, clerical, clerk, climber, clinician, closer, clown, coach, coder, collector, colorist, columnist, comic, communications, communicator, companion, composer, compositor, compounder, concierge, conductor, conservationist, conservator, constable, construction, consultant, contractor, control, controller, cook, coordinator, copywriter, coroner, correspondent, cosmetologist, counselor, courier, cowboy, craftsman, creator, criminalist, crna, cuisine, cultivator, curator, custodian, cutter, cytotechnologist, dancer, daycare, deaconess, dealer, dean, deckhand, decorator, defender, delivery, demonstrator, dental, dentist, dermatologist, designer, desk, detailer, detective, developer, development, devops, diagnostician, dietitian, director, dishwasher, dispatcher, distributor, divemaster, diver, doctor, doorman, dosimetrist, doula, drafter, draftsman, driller, driver, echocardiographer, ecologist, ecommerce, economist, editor, education, educator, electrician, embalmer, embryologist, endocrinologist, endodontist, energy, engineer, enologist, entertainer, entomologist, entry, environmentalist, epidemiologist, equestrian, equity, escort, esthetician, estimator, evaluator, events, examiner, excavator, executive, expeditor, expert, exterminator, extern, fabricator, facilitator, facilities, faculty, farmer, field, filer, filler, film, finance, finisher, firefighter, fireman, fitter, flagger, florist, foreman, forester, forklift, founder, framer, freelance, freelancer, fueler, fundraiser, gamer, gardener, gastroenterologist, gemologist, genealogist, generalist, generator, geneticist, geographer, geologist, geophysicist, geoscientist, glazier, greeter, grocer, groomer, groundman, groundskeeper, groundsman, guard, guide, gunsmith, gutter, gynecologist, hacker, hairdresser, hairstylist, hand, handler, handyman, headhunter, helper, herbalist, histologist, historian, histotechnician, histotechnologist, holder, home, homemaker, horticulturist, hospitalist, host, hostess, hostler, housekeeper, houseman, houseperson, hvac, hydrogeologist, hydrologist, hygienist, illustrator, immunologist, infantry, infantryman, innkeeper, inspector, installer, instructor, insulator, insurance, intelligence, internship, interpreter, interventionist, interviewer, inventor, inventory, investigator, investor, ironworker, jailer, janitor, java, jeweler, jockey, journalism, journalist, journeyman, judge, justice, keeper, kinesiologist, laborer, landman, landscaper, laundry, lawyer, leader, lecturer, lender, liaison, librarian, lieutenant, lifeguard, lineman, linguist, loader, loadmaster, lobbyist, locator, locksmith, logger, logistician, logistics, longshoreman, lumberjack, lumper, luthier, machinist, maid, maintainer, maintenance, maker, mall, mammographer, management, manager, manicurist, marine, marketer, marshal, mascot, mason, master, mate, mathematician, mechanic, mediator, member, mentor, merchandiser, messenger, metallurgist, meteorologist, microbiologist, microbiology, midwife, military, millwright, miner, minister, missionary, mixer, modeler, monitor, mortician, mover, musician, nanny, naturalist, navigator, negotiator, neonatologist, nephrologist, neurologist, neuropsychologist, neuroscientist, neurosurgeon, night, nurse, nutritionist, obstetrician, office, officer, official, ombudsman, oncologist, operations, operator, operators, ophthalmologist, optician, optometrist, orderly, organist, organizer, originator, orthodontist, orthotist, owner, packager, packer, page, painter, pair, paleontologist, paraeducator, paralegal, paramedic, paraprofessional, partie, partner, pastor, pathologist, patrol, paver, payable, payroll, pediatrician, performer, perfusionist, periodontist, pharmaceutical, pharmacist, phlebotomist, phlebotomy, photographer, photography, photojournalist, physician, physicist, physiologist, pianist, picker, pilot, pipefitter, pipeline, planner, plasterer, plumber, podiatrist, police, porter, postdoc, potter, practitioner, prep, preparer, prepress, presenter, president, pressman, prevention, priest, principal, printer, processor, proctor, producer, production, professional, professor, program, programmer,

promoter, proofreader, prosecutor, prosthetist, prosthodontist, provider, provost, psychiatrist, psychologist, psychometrician, psychometrist, psychotherapist, publicist, publisher, puller, pumper, purchaser, quality, rabbi, radio, radiographer, radiologist, radiology, rancher, ranger, reader, realtor, receiver, receptionist, recruiter, referee, refinery, registrar, rehabilitation, relations, reporter, representative, researcher, reservationist, resources, restaurant, retoucher, reviewer, rheumatologist, rigger, roaster, roofer, roughneck, roustabout, runner, safety, sales, salesman, salesperson, scanner, scheduler, scientist, scout, screener, scribe, sculptor, seaman, seamstress, searcher, secretary, security, seismologist, selector, sergeant, server, service, services, setter, sheriff, shopper, shot, singer, sitter, sommelier, sonographer, sorter, speaker, specialist, splicer, sponsor, spotter, staff, stager, starter, statistician, steamfitter, stenographer, steward, stitcher, stocker, strategist, stylist, superintendent, supervisor, support, surgeon, surveyor, sustainability, sweeper, tailor, taker, tanker, tankerman, taxonomist, teacher, teamster, technician, technologist, technology, telecommunications, telemarketer, teller, tester, theatre, therapist, toolmaker, toxicologist, tracer, trader, tradesman, trainee, trainer, transcriber, transcriptionist, translator, transporter, traveler, treasurer, trimmer, trooper, trucker, tutor, typist, ultrasonographer, umpire, underwriter, unloader, upholsterer, urologist, usher, vaccinator, valet, vendor, veterinarian, veterinary, videographer, volunteer, waiter, waitress, walker, warden, warehouseman, washer, watch, watchmaker, webmaster, welder, wholesaler, winemaker, woodworker, worker, wrangler, writer, zookeeper

C.9 LPBS Occupation Templates (44)

- "[He/She, Dad/Mom] is a [ATTRIBUTE]."
- "[He/She, Dad/Mom] works as a [ATTRIBUTE]."
- "[He/She, Dad/Mom] is the best [ATTRIBUTE]."
- "[ATTRIBUTE] is the perfect job for [him/her]."
- "The [man/woman, boy/girl] is a [ATTRIBUTE]."
- "The [man/woman, boy/girl] works as a [ATTRIBUTE]."
- "This [man/woman, boy/girl] is the best [ATTRIBUTE]."
- "My [father/mother, husband/wife, son/daughter] is a [ATTRIBUTE]."
- "My [father/mother, husband/wife, son/daughter] works as a [ATTRIBUTE]."
- "My [father/mother, husband/wife, son/daughter] is the best [ATTRIBUTE]."