Debiasing Methods in Natural Language Understanding Make Bias More Accessible

Michael Mendelson
michael.me@cs.technion.ac.il

Yonatan Belinkov[*]
belinkov@technion.ac.il

The Henry and Marilyn Taub Faculty of Computer Science
Technion – Israel Institute of Technology

Abstract

Model robustness to bias is often determined by the generalization on carefully designed out-of-distribution datasets. Recent debiasing methods in natural language understanding (NLU) improve performance on such datasets by pressuring models into making unbiased predictions. An underlying assumption behind such methods is that this also leads to the discovery of more robust features in the model's inner representations. We propose a general probing-based framework that allows for post-hoc interpretation of biases in language models, and use an information-theoretic approach to measure the extractability of certain biases from the model's representations. We experiment with several NLU datasets and known biases, and show that, counter-intuitively, the more a language model is pushed towards a debiased regime, the more bias is actually encoded in its inner representations.[1]

1 Introduction

State of the art neural language models such as BERT (Devlin et al., 2019) usually work by pre-training an encoder to learn universal word representations, and then fine-tuning it on some classification or regression task. From a robustness point of view, such pretrain-and-fine-tune pipelines are known to be prone to biases that are present in data (Gururangan et al., 2018; Poliak et al., 2018; Mc-Coy et al., 2019; Schuster et al., 2019). Various methods were proposed to mitigate such biases in a form of robust training, where a *bias model* is trained to capture the bias and then used to relax the predictions of a main model, so that it can focus less on biased examples and more on the "hard", more challenging examples (Clark et al., 2019; Mahabadi et al., 2020; Utama et al., 2020b; Sanh et al.,
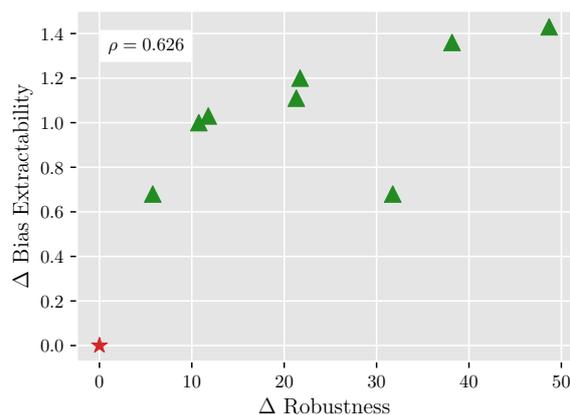

Figure 1: Amount of subsequence bias extracted from different language models vs. the robustness of models to the bias. Robustness is measured as improvement of the model on out-of-distribution examples, while extractability is measured as the improvement of the probe's ability to extract the bias from a debiased model, compared to the baseline.

2021, inter alia). Then, the resulting model is evaluated on out-of-distribution (o.o.d) data, in the form of challenge datasets containing "hard" examples that were deliberately constructed to be anti-biased. Examples of such datasets include HANS (McCoy et al., 2019) for natural language inference (NLI) and FEVER-Symmetric (Schuster et al., 2019) for fact verification. An underlying assumption behind this methodology is that better generalization out of distribution also means that the model learned more robust features. However, while evaluation using challenge datasets only relays information about the generalization of the model through predictions, it does not reveal what actually caused it and how the internal representations were affected.

To assess whether bias has been removed from the internal representations, we design probing tasks targeting several known biases: lexical overlap biases and negative word bias. While probing is usually concerned with simple linguistic properties such as part-of-speech tags (Belinkov and Glass,

---

[1] Our code and data are available at: https://github.com/technion-cs-nlp/bias-probing.

2019), we instead define probing tasks with the purpose of revealing bias in the representations. An example of such probing task is to predict whether a sentence-pair is lexically overlapping given only access to their joint representation—a classifier which is able to label the pair by this property consequently must use information about the bias which is encoded in the representation. We construct probing datasets for assessing bias in several natural language understanding (NLU) datasets. Lastly, we use information-theoretic probing (Voita and Titov, 2020) to analyze the extractability of bias from vanilla and debiased models using the probing classifier.

We conduct experiments on two NLI datasets and one fact verification dataset across a variety of debiasing methods and bias types, and test whether the bias removal is as successful as o.o.d evaluation suggests. Surprisingly, we discover that making models robust from the perspective of the downstream task, causes the inner representations to encode more of the information about the specific bias in question. Figure 1 shows an example of this trend in NLI, where as robustness of the fine-tuned model to biased predictions increases, so does the ability of the probing classifier to extract bias.

To summarize, we make the following contributions:

- We present a general probing-based framework to measure extractability of bias from inner model representations.

- We use this framework to construct several new probing tasks based on well-studied dataset biases in NLU tasks.

- We show that pressuring a model into making unbiased predictions actually makes biased features more extractable from the model representations.

2 Related Work

2.1 Dataset Biases

Deep neural models are prone to shortcut learning (Geirhos et al., 2020), by discovering and using idiosyncratic biases, heuristics, and statistical cues in the data. For example, Poliak et al. (2018) showed that the Stanford natural language inference dataset (SNLI; Bowman et al. 2015) contains "give-away" words, i.e., words w which have a high value of $p(l \mid w)$ w.r.t a given label l. They noticed that 4 out of the 10 words with the high-

est $p(\text{contradiction} \mid w)$ are universal negation words,[2] suggesting that negation is strongly correlated with contradiction in the data. These clues appeared in the hypothesis side, making them a kind of hypothesis-only bias, where a classifier receiving as input only the hypothesis is able to correctly predict the label (Poliak et al., 2018; Gururangan et al., 2018). A similar type of bias, known as claim-only bias, is found in the FEVER fact verification dataset (Thorne et al., 2018), and was also associated with a strong correlation of negation words with the labels in the dataset (Schuster et al., 2019). Another kind of bias is the association of entailment with cases of lexical overlap between the premise and hypothesis. This bias leads to poor performance of models on the HANS challenge dataset (McCoy et al., 2019), where all samples contain lexical overlap and non-entailed samples are formed such that the bias does not entail the label. This suggests that models rely on features that are cues for lexical overlap bias when predicting the entailment of premise–hypothesis pairs.

2.2 Bias Mitigation and Robustness

Recent work on bias mitigation attempts to create more robust models by training a combination model, based on the main model. The main model, parameterized by θ_m, is a non-robust language model. The bias model, parameterized by θ_b, is a *weak model* whose purpose is to model the biases during training, by minimizing a loss $\mathcal{L}_b$. The objective of the combination model is to minimize a combined loss function $\mathcal{L}_c(\theta_m, \theta_b)$, such that the main model leverages knowledge about bias in data, obtained using the weak model. This pipeline is general, and it allows models to be trained either end-to-end, or step-by-step by first training the bias model and then using its predictions to robustly train the main model. Recent papers show that such techniques are effective when evaluated on challenge datasets specifically designed to target known biases and hard examples (He et al., 2019; Clark et al., 2019; Utama et al., 2020b,a; Sanh et al., 2021; Mahabadi et al., 2020). However, this approach does not ensure that the model indeed learns more robust features, nor does it shed light on exactly *how* the feature detectors react to this change, and how the bias is represented in the model.

---

[2]nobody, alone, no, empty.

2.3 Probing

Probing was somewhat successfully used to analyze sentence embeddings and to show that such models capture surface features such as sentence length, word content, and the order of words (Adi et al., 2017), or various syntactic and semantic features (Conneau et al., 2018); see Belinkov and Glass (2019) for a survey. In contrast, we focus our analysis on *biased* features, and employ advances in probing methodology to analyze two kinds of bias—lexical overlap and negation bias. Designing probes to accurately interpret the desired behavior is not trivial and measuring their accuracy is insufficient, since the probing classifiers are prone to memorization and bias as well (Hewitt and Liang, 2019), among other shortcomings (Belinkov, 2021). Recently, Voita and Titov (2020) presented an information-theoretic approach for evaluating probing classifiers, which accounts for the complexity of the probing classifier by measuring its minimum description length (MDL). MDL measures how efficiently a model can extract information about the labels from the inputs, and we use it as a measure of extractability of certain biases from model representations.

3 Methods

We lay down a general framework for interpreting bias in inner model representations. Given a model $f_\theta : X \to Y$ with learnable parameters θ, we assume that it can be decoupled into two stages:

- A representation layer (or multiple layers) with learnable parameters θ_1, which we denote $\mathcal{R}_{\theta_1} : X \to Z$, maps samples from the input space to a latent space Z, the "representation".

- A classification layer with learnable parameters θ_2, which we denote $\mathcal{F}_{\theta_2} : Z \to Y$, maps the latent representations to the final output.

We can thus re-define our classifier as

$$f_\theta (x) \triangleq \mathcal{F}_{\theta_2} \left(\mathcal{R}_{\theta_1} (x) \right). \qquad (1)$$

For example, in NLI we assume that data samples are given as sentence pairs $x = (p, h)$ where p is a premise and h is a hypothesis. $\mathcal{R}(p, h)$ is the joint representation of the two, and this representation is then used by $\mathcal{F}$ to produce a prediction.

In this work, we compare baseline models fine-tuned on some down-stream task to models debiased during the fine-tuning step. We produce representations from both types of models and measure the extractability of bias using a probing classifier. Our probing tasks are defined in terms of "bias-revealing" properties, which are based on a-priori knowledge of the bias in question, and are able to distinguish between biased and unbiased samples from the original dataset. We next describe how to construct such probing tasks and appropriate datasets.

3.1 Probing Tasks and Datasets

We define a probing classifier as a classifier $g_\Psi : Z \to Y_P$ with learnable parameters Ψ, which maps inputs from a latent representation space Z to a probing property space Y_P, where $P : X \to Y_P$ is some real property of the original input, which we call the *probing property*. Next, we define a *probing dataset* for each probing task:

$$\mathcal{D}_P = \{ (\mathcal{R}_\theta (x), P (x)) \mid x \in X \}. \qquad (2)$$

Lastly, we train the probing classifier on the constructed dataset and evaluate its performance on the probing task. We introduce two new probing tasks that target the well researched types of bias present in several datasets: lexical bias and negative word bias. For presentation purposes, consider the NLI task, where data samples are given as sentence pairs $x = (p, h)$ where p is a premise and h is a hypothesis. The extension to fact verification and other pair relationship classification tasks is straightforward.

NegWords To analyze negation bias in NLI and fact verification, we define a list of negative words V[3] and a sentence pair property

$$P_{\text{neg}}^V (p, h) = \mathbb{1} \left[V \cap h \neq \emptyset \right]. \qquad (3)$$

That is, an example is positive if its hypothesis (in the case of NLI) or claim (in the case of fact verification) contains at least one negative word from the list. This method poses some limitations: For example, we do not consider double negatives in the hypothesis that affect its meaning, or the presence of negation in both premise and hypothesis. However, our construction is consistent with prior findings on negation bias (Gururangan et al., 2018; Poliak et al., 2018; Schuster et al., 2019).

---

[3]In our experiments we use $V = \{$no, not, nobody, never, nothing, none, empty, neither, cannot$\} \cup \{$Words that end with *n't*$\}$ for a total of $|V| = 27$ words.

Overlap/Subsequence Based on the analysis of McCoy et al. (2019), we define a class of probing tasks for identifying the different lexical heuristics in NLI. We focus on lexical overlap and subsequences[4] and define two sentence pair properties:

$$P_{\text{lex}}(p, h) = \mathbb{1}\left[h \subseteq p\right], \quad (4)$$

where an example is positive if all the hypothesis words are found in the premise (regardless of word order), and

$$P_{\text{sub}}(p, h) = \mathbb{1}\left[h \text{ is a subsequence of } p\right], \quad (5)$$

where an example is positive if the hypothesis is a subsequence of the premise.

3.2 Data Processing

To alleviate issues of data balancing, we take the following steps when processing the probing datasets: First, we identify all the biased samples in a given dataset, according to the probing property. Since in all our datasets the positive class (biased samples) is the minority class, we subsample the same amount of samples from the remaining subset (the majority class). We end up with a balanced probing dataset. This ensures that when splitting the data during online code training, and when measuring performance on the entire dataset, the process is unaffected by the bias *evidence*, that is, the amount of bias in the original dataset.

The probing datasets are constructed from three base NLU datasets: SNLI (Bowman et al., 2015), MultiNLI (Williams et al., 2018) and FEVER (Thorne et al., 2018), following the original train/validation/test splits.[5] Inspired by previous work on biases in NLU datasets (Section 2), we construct **NegWords** probing datasets from all three base NLU datasets and **Overlap/Subsequence** probing datasets from SNLI and MNLI. The dataset statistics are presented in Table 1.

3.3 Evaluation

We use a linear probe across all experiments. We evaluate both the probe's accuracy and its minimum description length (MDL; Voita and Titov 2020), to measure bias extractability. Formally, given a dataset $\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ and a probabilistic model $p_\theta(y \mid x)$, the description length

---

[4]We exclude the constituency heuristic since it is not frequent enough in MNLI to construct a probing dataset.

[5]FEVER does not provide a test set, and we therefore report results on the validation set, and do not perform any type of hyperparameter tuning.

Task	Dataset	Train	Valid	Test
NegWords	SNLI	25104	484	456
	MNLI	126232	3180	3246
	FEVER	19874	2180	–
Overlap	SNLI	35388	734	732
	MNLI	18542	518	464
Sub.	SNLI	4438	234	226
	MNLI	5432	202	154

Table 1: Number of samples in all probing datasets created from the different base datasets.

of the model is defined as the number of bits required to transmit the labels $Y = (y_1, \ldots, y_n)$, given $X = (x_1, \ldots, x_n)$. We estimate MDL using Voita and Titov's *online coding*, and denote the result L_{online}. Given a uniform distribution over the K labels, we get $L_{\text{unif}} = |\mathcal{D}| \log K$. Thus, the *compression* is defined as $\mathcal{C} = \frac{L_{\text{unif}}}{L_{\text{online}}}$ and it holds that $1 \leq \mathcal{C} \leq \mathcal{C}^*$ where $\mathcal{C}^*$ is the compression given by a perfect model. We interpret a lower MDL score (and consequently, a higher compression score) to mean that the probing property is more extractable from the model representation. The hyperparameters we use in the evaluation process are outlined in Appendix A.1.

3.4 Debiasing Methods

To deploy our framework in the context of robustness to bias, we examine several proposed strategies for debiasing NLU models. In all cases, a weak learner models the bias and is combined with a main model to produce less biased predictions.

We note that there are three different criteria for controlling the debiasing strategy: (1) Models may be trained *end-to-end* by propagating errors to the weak learner as well as the main model (Mahabadi et al., 2020) or in a *pipeline*, where the weak learner is trained first and frozen, such that only its predictions are used to tune the combination loss (He et al., 2019; Clark et al., 2019; Sanh et al., 2021; Utama et al., 2020a). (2) The bias model can accept the bias either *explicitly* (by accepting only a set of predefined biased features x^b, as in most work) or *implicitly*, by training it in a weak setting: Sanh et al. (2021) train a small model (TinyBERT; Turc et al. 2019) and rely on its limited size to adopt biased representations, while Utama et al. (2020b) train a BERT-size model on a small subset of the training set, to allow it to capture weaker features

of the data. (3) The *objective function* by which the main and bias model are combined can vary. Below we describe three common objective functions. We test different combinations of all strategies where they are feasible, resulting in a wide array of debiased models.

3.4.1 Debiasing Objectives

Debiased Focal Loss (DFL) Focal loss was first proposed by Lin et al. (2017) to encourage a classifier to focus on the harder examples, for which the model is less confident. This is achieved by weighing standard cross-entropy with $(1 - p_m)^\gamma$, where p_m is the class probability and γ is the focusing parameter. Mahabadi et al. (2020) propose DFL, where the weighting is achieved by a bias-only model's class probability p_b and the loss becomes:

$$-\frac{1}{N} \sum_{i=1}^{N} (1 - p_b)^\gamma \log p_m. \qquad (6)$$

We re-implement their model with two bias-only models: a hypothesis-only model and a lexical bias model that uses the same input features as Mahabadi et al. (2020), outlined in Appendix A.2

Product of Experts (PoE) Product of experts (PoE) was first proposed by Hinton (2000) as a method for training ensembles of models that are experts at specific sub-spaces of the entire distribution space. Each model can focus on an "area of expertise" and their multiplied predictions form the combination model. This idea was utilized in several studies (He et al., 2019; Clark et al., 2019; Mahabadi et al., 2020; Sanh et al., 2021) to train a combination of models where the experts are weak models. The combination model output becomes

$$\mathcal{F}_c(x) = \text{softmax}(\log p_b + \log p_m), \qquad (7)$$

and is trained with standard cross-entropy.

Confidence Regularization (ConfReg) In this method, proposed by Utama et al. (2020a), a bias-only/weak model and a teacher model are first independently trained on the target dataset. Then, the predictions of the teacher model are downweighed by the predictions of the weak model. The weighted loss is then used to distill knowledge (Hinton et al., 2015) to a new main model, parameterized in the same way as the teacher model (this is known as *self distillaion*). We note that ConfReg cannot be easily trained in an end-to-end setting, because it relies on an already trained teacher model to down-weigh the predictions.

4 Experiments

4.1 Datasets

We use three English NLU datasets: SNLI, MNLI and FEVER. They are used both for training baseline and debiased models, and to create probing datasets for our tasks, as described in Section 3.2.[6]

SNLI The SNLI dataset contains around 570k premise-hypothesis pairs with three possible labels: **entailment** if the premise entails the hypothesis, **contradiction** if the premise contradicts the hypothesis, or **neutral** if neither hold. We evaluate on the hard subset (Gururangan et al., 2018), designed to have fewer hypothesis-only biases.

MNLI The MNLI dataset is a multi-genre variant of SNLI which contains around 430k premise-hypothesis pairs. We evaluate on a hard subset of the dev matched set, provided by Mahabadi et al. (2020), which was created by taking examples that a hypothesis-only classifier failed to classify.

FEVER The Fact Extraction and VERification (FEVER) dataset contains around 180k pairs of claim–evidence pairs, where the task is to predict one of three labels: either the evidence **supports** or **refutes** the claim, or there is **not enough information**. We evaluate on **FEVER-Symmetric**, which was designed such that it cannot be predicted by a claim-only classifier (Schuster et al., 2019).

4.2 Models

We test different models based on BERT, by removing the classification head and using the pooled representation of the `[CLS]` token as input to our probes. In settings where previous work compared in-distribution and o.o.d performance, we use hyperparameters which are known to work well for the task and dataset. For new settings which were not reported in previous work, we sweep for the best hyperparameters based on the in-distribution accuracy on the validation set.[7] All hyperparameters are available in Appendix A.3. We train all models with five random seeds and report means

---

[6] Our probing tasks contain examples from all original labels of the datasets. A reviewer pointed out that one can look at probing datasets where examples are drawn only from a specific down-stream label, but our experiments found that splitting per label does not reveal different trends than those we observe here.

[7] In our experiments, some methods did not converge, notably PoE and DFL using a model with subset sampling. This method was used to train ConfReg models and is likely much more sensitive to selection of the weak model.

Bias	Model	Overlap			Subsequence		
		$\mathcal{C}$	Acc.	HANS$^-$	$\mathcal{C}$	Acc.	HANS$^-$
	Random	1.4 ± 0.0	59.7 ± 2.7	–	1.4 ± 0.0	64.9 ± 4.5	–
	Pretrained	1.9 ± 0.0	77.4 ± 0.2	–	1.9 ± 0.0	80.9 ± 0.6	–
	Base	3.2 ± 0.2	88.8 ± 1.2	38.9 ± 18.8	3.2 ± 0.3	91.3 ± 3.6	6.5 ± 3.0
Explicit	DFL$_{e2e}$	4.0 ± 0.5	92.6 ± 1.3	67.4 ± 9.7	4.4 ± 0.5	95.1 ± 2.6	28.4 ± 6.6
	PoE$_{e2e}$	4.0 ± 0.5	91.7 ± 0.7	65.3 ± 4.8	4.2 ± 0.5	92.5 ± 0.7	17.4 ± 1.8
Subset	ConfReg	4.6 ± 0.5	92.3 ± 1.6	53.2 ± 14.2	4.3 ± 0.4	93.4 ± 1.7	18.4 ± 5.9
	DFL	4.1 ± 0.1	92.2 ± 0.7	57.1 ± 13.0	3.9 ± 0.2	93.5 ± 2.0	38.4 ± 16.4
Tiny	DFL	4.8 ± 0.3	$\mathbf{93.6 \pm 1.1}$	$\mathbf{75.3 \pm 4.8}$	4.6 ± 0.4	94.7 ± 1.9	45.9 ± 6.9
	DFL$_{e2e}$	$\mathbf{4.9 \pm 0.3}$	93.0 ± 1.0	74.0 ± 5.8	$\mathbf{4.7 \pm 0.2}$	$\mathbf{95.1 \pm 1.4}$	$\mathbf{57.6 \pm 9.6}$
	PoE	3.6 ± 0.3	90.9 ± 1.1	63.5 ± 5.5	3.9 ± 0.5	93.3 ± 1.3	13.2 ± 4.2
	PoE$_{e2e}$	4.2 ± 0.1	92.0 ± 0.8	73.1 ± 6.6	4.3 ± 0.2	94.3 ± 2.3	27.2 ± 5.2

Table 2: Results of probing for Overlap and Subsequence on MNLI. $\mathcal{C}$ is the compression of the probing classifier and *Acc* is the accuracy. HANS$^-$ identifies the performance of the original model on the relevant subset of non-entailed samples in HANS: (1) the lexical overlap subset for **Overlap**, (2) the subsequence subset for **Subsequence**. We report results for models with different bias models: (1) explicit bias-only model with lexical overlap features, (2) implicit bias model with subsampling (Subset), and (3) implicit TinyBERT bias model (Tiny).

and standard deviations, to account for known variability of fine-tuned models, especially when evaluated out of distribution (McCoy et al., 2020).

We reimplement all debiasing methods in a unified codebase to facilitate a fair comparison. Training details are available in Appendix A.4.

Baselines We use the standard base BERT implementation of Wolf et al. (2020). We take the pretrained model without further fine-tuning on any downstream task (denoted as Pretrained) and we also fine-tune the model on the target dataset (Base). To obtain a lower bound on the performance of these models, we take the same model and randomly initialize its weights (Random).

5 Results

In this section, we first report our main finding—the correlation between the robustness of models. We then analyze each bias type and dataset in a more fine-grained manner.

Table 3 shows the Pearson correlations (ρ) between robustness and bias extractability. Robustness is measured as the difference between the performance of a debiased model on a relevant o.o.d dataset and that of a baseline model. Higher values mean that the debiased model is more robust. Bias extractability is measured as the compression score using a probing classifier designed to target the bias. In all but one case, we find positive correlations,

indicating that the more successful a method is in debiasing model predictions, the more it makes the bias accessible in the inner representations.

The only exception is NegWords bias on MNLI, where we report a negative correlation. As we analyze below, in this case some models do not improve on o.o.d data, but their compression still increases. This suggests that even though various debiasing methods are not always successful on different datasets and bias types, they still make bias more accessible in the representations.

Bias	Dataset	M	ρ
NegWords	SNLI	6	0.757
	MNLI	7	-0.257
	FEVER	7	0.289
Overlap	SNLI	7	0.752
	MNLI	8	0.358
Sub.	SNLI	7	0.672
	MNLI	8	0.626

Table 3: Correlation between bias extractability and robustness in various bias types and datasets. M = number of models over which the correlation is measured.

5.1 Lexical Bias

MNLI Table 2 shows results for the Overlap/Subsequence probing tasks, on MNLI. For each model, we report compression ($\mathcal{C}$) and accuracy of

		Overlap			Subsequence		
Bias	Model	$\mathcal{C}$	Acc.	HANS$^-$	$\mathcal{C}$	Acc.	HANS$^-$
	Random	1.4 ± 0.0	61.1 ± 2.1	–	1.4 ± 0.0	55.8 ± 3.9	–
	Pretrained	2.2 ± 0.0	83.0 ± 0.1	–	2.2 ± 0.0	81.2 ± 0.2	–
	Base	4.6 ± 0.4	93.8 ± 1.1	48.4 ± 6.3	5.4 ± 0.9	94.7 ± 2.3	2.4 ± 1.1
Explicit	DFL$_{e2e}$	$\mathbf{5.8 \pm 0.6}$	$\mathbf{94.6 \pm 0.3}$	69.1 ± 9.7	$\mathbf{6.7 \pm 0.8}$	$\mathbf{95.2 \pm 2.1}$	$\mathbf{21.0 \pm 18.9}$
	PoE$_{e2e}$	4.9 ± 0.3	93.8 ± 0.7	65.0 ± 10.9	5.7 ± 0.5	95.0 ± 0.9	7.9 ± 4.4
Subset	ConfReg	4.2 ± 0.3	93.4 ± 0.6	62.0 ± 10.3	4.4 ± 0.4	93.0 ± 0.9	14.9 ± 6.0
Tiny	DFL	4.1 ± 0.5	92.6 ± 1.6	55.7 ± 8.7	4.5 ± 1.0	91.8 ± 2.2	6.9 ± 4.3
	DFL$_{e2e}$	5.0 ± 0.3	94.2 ± 0.7	69.4 ± 8.2	5.6 ± 0.6	94.7 ± 1.5	13.6 ± 6.7
	PoE	5.0 ± 0.4	93.9 ± 0.7	64.6 ± 9.3	6.0 ± 0.6	94.5 ± 0.9	13.5 ± 4.8
	PoE$_{e2e}$	4.9 ± 0.3	94.2 ± 0.4	$\mathbf{70.8 \pm 5.1}$	5.7 ± 0.7	94.7 ± 1.6	15.6 ± 6.8

Table 4: Results of probing for lexical bias on SNLI. The notation here stays consistent with Table 2.

the probe and the performance of the model on anti-biased (non-entailed) samples from the relevant subset of HANS attributed to the lexical overlap heuristic (HANS$^-$ column).

All debiasing methods improve the o.o.d generalization (performance on HANS$^-$) compared to the base model, consistent with prior work. All debiasing methods also lead to models with more extractable bias, as demonstrated by higher compression values. The base model already exhibits higher compression than a random model or a pretrained model, indicating that fine-tuning makes bias more extractable from the inner representation. However, fine-tuning with any debiasing method makes this bias even more extractable.

In fact, as performance on the anti-biased examples from the HANS subset increases, so does the compression of the probe; Figure 1 shows an example of this trend in the subsequence case. DFL with implicit bias from the TinyBERT model (trained either end-to-end or in a pipeline) has the highest compression values, as well as the biggest improvement out of distribution.

SNLI Table 4 shows results for the Overlap and Subsequence probing tasks. All debiasing methods lead to improved o.o.d performance, as expected. Compression of the random and pretrained baselines remains very close, with most of the bias being made more extractable in the representations of the fine-tuned baseline (Base). Most of the debiased models still largely surpass the baseline for compression and probing accuracy, indicating that they make bias more extractable. ConfReg and

Bias	Model	$\mathcal{C}$	Symmetric
	Random	1.37 ± 0.00	–
	Pretrained	1.64 ± 0.03	–
	Base	2.97 ± 0.10	56.0 ± 2.0
Claim	DFL$_{e2e}$	3.04 ± 0.08	62.1 ± 1.8
	PoE$_{e2e}$	3.00 ± 0.05	61.9 ± 1.6
Subset	ConfReg	3.03 ± 0.04	56.2 ± 2.0
Tiny	DFL	$\mathbf{3.31 \pm 0.09}$	$\mathbf{62.2 \pm 3.9}$
	DFL$_{e2e}$	3.16 ± 0.07	60.5 ± 2.5
	PoE	3.12 ± 0.05	61.0 ± 3.6
	PoE$_{e2e}$	3.06 ± 0.06	61.4 ± 3.2

Table 5: Results of probing for **NegWords** on FEVER. Symmetric is the o.o.d set by Schuster et al. (2019), which is designed such that a claim-only classifier cannot achieve higher-than-guess performance on it. Probing accuracy is reported in Appendix A.5.

DFL with a fine-tuned TinyBERT are exceptions; they do not exhibit higher compression than the baseline, but still improve out of distribution.

5.2 Negative Word Bias

FEVER Table 5 shows the results for the NegWords task on FEVER. All models improve on FEVER-Symmetric compared to the baseline (Base), indicating that they are less biased in their predictions. Conversely, when probed for the bias, all models achieve higher compression compared to the baseline and outperform it in terms of probing accuracy. That is, this bias is more extractable in the debiased models than in the baseline model. As a point of reference, the compression of the

random model is smallest, closely followed by the pre-trained model. Any fine-tuning leads to significantly larger compression scores. These trends are consistent with the Overlap/Sub. results. The best model in terms of o.o.d accuracy is DFL with an implicit TinyBERT bias model. We also see that bias is most extractable in this model, compared to the baseline. While previous work used statistical tools to show that the REFUTES label is spuriously correlated with negative bigrams (Schuster et al., 2019), we reveal that this information is preserved and even amplified in the model when an attempt is made to make the predictions less reliant on it.

SNLI In this case, all models perform better or as well as the baseline model when evaluated on the hard subset, yet the compression values of all models significantly surpass the baseline. While any form of debiasing makes bias more available in the representations, it does not necessarily lead to an improvement on the o.o.d set. Models with a hypothesis-only model perform best out of distribution, and also expose the most bias. Similarly to the results on FEVER, the compression of the random and pretrained models is significantly lower and close to each other, with most of the bias being made available by fine-tuning the model (Base). Table 7 in Appendix A.5 provides the full results.

MNLI Compression results are much closer to the fine-tuned baseline, but all debiased models still contain more information about negation words. This is on-par with previous results that anaylzed the statistical correlation of such negation words to the CONTRADICTION label (Gururangan et al., 2018; Poliak et al., 2018), and we show that not only does the correlation exist in the data, but attempts to remove such evidence result in more extractability. Still, most growth in compression compared to the random and pretrained models is attributed to the fine-tuning process itself (without debiasing). Interestingly, some of the models do not improve the performance on the hard test set, but their compression still increases, suggesting that the more accessible bias can also be decoupled from the predictions of the model. Table 8 in Appendix A.5 provides the full results.

5.3 Varying the Debiasing Effect

So far we evaluated the effect of debiasing on bias extractability across debiasing methods. To evaluate this effect within the same method, we analyze the effect of stronger debiasing in the DFL method,

by increasing the "focusing parameter" γ (Eq. 6). We test our probing tasks on models trained with increasing values of $\gamma \in \{1, 2, 3, 4\}$. Figure 2 shows the results for the Overlap/Subsequence tasks. As we increase γ, the extractability of bias from the model's representations increases. This is consistent with our main results.
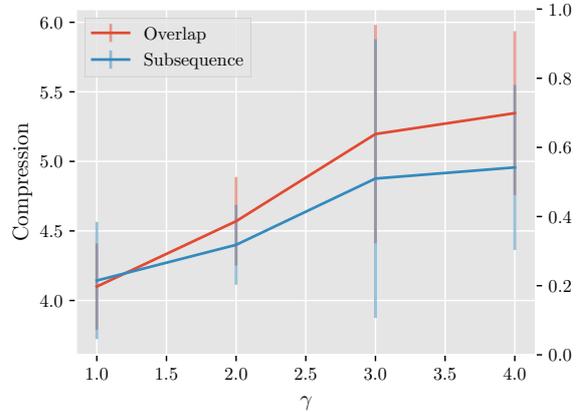

Figure 2: Compression of a DFL model with an implicit bias model on Overlap/Subsequence probing tasks vs. the focusing parameter γ, for MNLI. As γ increases, the bias becomes more extractable.

5.4 Linguistic Information in Debiased Models

Following the main results, a useful question to ask is whether debiased models also tend to learn useful linguistic information more broadly, which may explain the noticeable increase in performance out of distribution.[8] To test this, we take our models trained for NLI on the MNLI dataset and apply the SentEval probing tasks (Conneau et al., 2018), which test ten different linguistic properties in model representations. We exclude the word content (WC) task, because it is a 1000-way classification problem and takes substantially more time to train with an MDL probe. Table 6 shows the average results for all debiased models and the remaining nine tasks, compared to our three baselines (random, pretrained, fine-tuned). First, we notice that for 8/9 tasks, compression of the model decreases when it is fine-tuned, compared to the pretrained model. This can be explained by the close connection between the linguistic phenomena and the masked language modelling (MLM) objective, compared to fine-tuning on NLI. Furthermore, on average, debiased models do not decrease

---

[8]We thank a reviewer for pointing out this idea.

	$\mathcal{C}$				
	Random	Pretrained	Baseline	Average	Accuracy
BShift	1.39 ± 0.01	2.41 ± 0.0	1.61 ± 0.01	1.67 ± 0.03	51.6 ± 0.57
CoordInv	1.37 ± 0.01	1.58 ± 0.0	1.48 ± 0.01	1.5 ± 0.02	59.0 ± 1.74
ObjNum	1.4 ± 0.01	1.79 ± 0.0	1.77 ± 0.02	1.86 ± 0.04	73.8 ± 1.0
SOMO	1.37 ± 0.0	1.48 ± 0.0	1.44 ± 0.01	1.45 ± 0.01	58.7 ± 0.5
Tense	1.48 ± 0.01	3.05 ± 0.0	2.38 ± 0.12	2.52 ± 0.1	83.8 ± 1.25
SentLen	3.0 ± 0.16	2.19 ± 0.0	1.49 ± 1.2	2.24 ± 0.06	50.8 ± 0.8
SubjNum	1.39 ± 0.0	2.11 ± 0.0	1.83 ± 0.03	1.96 ± 0.05	76.2 ± 0.92
TopConst	1.68 ± 0.0	2.82 ± 0.0	2.41 ± 0.06	2.47 ± 0.14	51.9 ± 3.28
TreeDepth	1.48 ± 0.0	1.55 ± 0.0	1.53 ± 0.01	1.56 ± 0.01	25.6 ± 0.6

Table 6: Average accuracy and compression scores for debiased models and baselines, when probed for the SentEval tasks (Conneau et al., 2018). Random is the randomly initialized model, Pretrained is the pretrained model without fine-tuning, and Base is the fine-tuned model. Accuracy and Average denote the average accuracy and compression score of M debiased models trained on MNLI ($M = 8$).

in compression compared to the fine-tuned model, but the differences are very subtle and generally within standard deviation bounds. This suggests that while debiasing does not make linguistic information measured in these probing tasks less extractable, it also does not substantially amplify it, as opposed to extractability of bias information.

6 Discussion and Conclusion

All of our experiments tested model-based debiasing, where a weak learner is used to capture biased features and discourage their use in model predictions. We discover that for both explicit and implicit modeling of the bias, this method exposes the biased features in the representation. When we fix the model and change the effect of debiasing (through the "focusing parameter" of DFL), we observe the same trend, where stronger bias mitigation leads to higher extractability of the modelled bias. Based on our results, we stipulate that while current debiasing methods are good at making model predictions less biased, they are a bad proxy for learning unbiased text representations. The increased extractability of bias from the representations is not necessarily a bad trait: For example, the NegWords task does not reveal more granular semantics of negation, which may be useful for the generalization of the model. By probing for linguistic properties using the SentEval tasks, we also observe that debiased models do not make linguistic information less extractable, which can also contribute to their improvement in performance. We argue that future research should look for more

interpretable methods for debiasing language models, and consider the problem of finding robust, bias-free feature detectors.

Another domain where this finding may be alarming is social bias. Previous studies show that word vectors contain social bias (Caliskan et al., 2017), and that debiasing them does not necessarily remove this information (Gonen and Goldberg, 2019). Our work shows that debiasing sometimes increases the information available about bias in the representations, albeit in the context of dataset bias rather than social bias.

Our work shows that *unbiased predictions $\implies$ biased representations*. We speculate that there exists a proxy for the language model that removes bias information from the representations and consequently improves the generalization of predictions out of distribution. Future work could focus on methods that are both representation-robust and prediction-robust w.r.t various biases. Finding such methods can help alleviate leakage of bias from data to the model's representations, without sacrificing the in-distribution performance.

Acknowledgments

References

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. In *International Conference on Learning Representations*.

Yonatan Belinkov. 2021. Probing classifiers: Promises, shortcomings, and advances. *arXiv preprint arXiv:2102.12452*.

Yonatan Belinkov and James Glass. 2019. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics (TACL)*, 7:49–72.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

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

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 2019. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4069–4082, Hong Kong, China. Association for Computational Linguistics.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

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

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673.

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. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 609–614, Minneapolis, Minnesota. Association for Computational Linguistics.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

He He, Sheng Zha, and Haohan Wang. 2019. Unlearn dataset bias in natural language inference by fitting the residual. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 132–142, Hong Kong, China. Association for Computational Linguistics.

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

Geoffrey Hinton. 2000. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14:2002.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007.

Karimi Rabeeh Mahabadi, Yonatan Belinkov, and James Henderson. 2020. End-to-end bias mitigation by modelling biases in corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8706–8716, Online. Association for Computational Linguistics.

R. Thomas McCoy, Junghyun Min, and Tal Linzen. 2020. BERTs of a feather do not generalize together: Large variability in generalization across models with similar test set performance. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 217–227, Online. Association for Computational Linguistics.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, Louisiana. Association for Computational Linguistics.

Victor Sanh, Thomas Wolf, Yonatan Belinkov, and Alexander M Rush. 2021. Learning from others' mistakes: Avoiding dataset biases without modeling them. In *International Conference on Learning Representations*.

Tal Schuster, Darsh Shah, Yun Jie Serene Yeo, Daniel Roberto Filizzola Ortiz, Enrico Santus, and Regina Barzilay. 2019. Towards debiasing fact verification models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3419–3425, Hong Kong, China. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: On the importance of pre-training compact models.

Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. 2020a. Mind the trade-off: Debiasing NLU models without degrading the in-distribution performance. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8717–8729, Online. Association for Computational Linguistics.

Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. 2020b. Towards debiasing NLU models from unknown biases. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7597–7610, Online. Association for Computational Linguistics.

Elena Voita and Ivan Titov. 2020. Information-theoretic probing with minimum description length. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*,

pages 183–196, Online. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

A Appendix

A.1 Online Code Evaluation

Following Voita and Titov (2020), we evaluate our models using an online code probe, with timestamps [2.0, 3.0, 4.4, 6.5, 9.5, 14.0, 21.0, 31.0, 45.7, 67.6, 100] (Each timestamp corresponds to a percentage of the samples in the training dataset). We use a slightly different scale than Voita and Titov (2020), to account for the smaller datasets and the resulting instability in the first fractions of training. The last timestamp is used to train the probe on the full training dataset, and it is then evaluated for accuracy on the entire test set. During all training phases, we employ early-stopping when the validation accuracy does not improve over four epochs, with a tolerance of 10^{-3}.

A.2 Bias-only Models

For the lexical bias-only model, we use the following features as bias input features: 1) Whether all words in the hypothesis are included in the premise; 2) If the hypothesis is the contiguous subsequence of the premise; 3) If the hypothesis is a subtree in the premise's parse tree; 4) The number of tokens shared between premise and hypothesis normalized by the number of tokens in the premise, and 5) The cosine similarity between premise and hypothesis's pooled token representations from BERT followed by min, mean, and max-pooling. Following Mahabadi et al., we also give equal weights to neutral and contradiction labels (by calculating a weighted cross-entropy loss) to encourage the model towards biased predictions.

A.3 Hyperparameters

ConfReg We train all models for five epochs and use the same hyperparameters as in Utama et al. (2020b): 2000 samples for the weak learner sub-sampling, a batch size of 32, learning rate of $5 \cdot 10^{-5}$, a weight decay of 0.01 and a linear scheduler for modulating the learning rate with a 10% warm-up proportion. For training FEVER, we set a learning rate of $2 \cdot 10^{-5}$ and sub-sample 500 samples. For SNLI we use the same parameters as in MNLI, but we sub-sample 3 000 samples to account for the larger dataset, and make sure that the weak model still follows the constraints: at least 90% of the predictions on the sampled training set fall within the 0.9 probability bin, and the weak learner achieves more than 60% accuracy on the entire training set.

DFL and PoE We train all models for three epochs on MNLI and SNLI with a batch size of 32, learning rate of $5 \cdot 10^{-5}$, a weight decay of 0.0 and a linear scheduler for modulating the learning rate with a 10% warm-up proportion. We choose $\gamma = 2.0$ for most of the DFL models. Exceptions are made for DFL with the subsampled bias model and end-to-end DFL with a TinyBERT bias model, where we sweep $\gamma \in \{1.0, 2.0\}$ and choose $\gamma = 1.0$ based on the highest validation accuracy (in-distribution). Another exception is made for FEVER, where we set the learning rate at $2 \cdot 10^{-5}$ to be consistent with previous work.

A.4 Training Details

To train all models, we have used single instances of NVIDIA GeForce RTX 2080 Ti, with an average training time of 1–7 hours. Models where the weak learner is frozen have 110M parameters, as in the base BERT model. TinyBERT models have 4.4M parameters (Turc et al., 2019) and any combination of a weak model and a main model is straightforward to calculate.

A.5 Additional Results

Table 7 summarizes the results for NegWords bias on the SNLI dataset, and Table 8 summarizes the results on MNLI. Table 9 shows the full results for NegWords on FEVER, including probing accuracy.

Bias	Model	$\mathcal{C}$	Acc.	Hard
	Random	1.47 ± 0.0	59.8 ± 2.4	–
	Pretrained	2.01 ± 0.0	76.1 ± 0.0	–
	Base	3.48 ± 0.3	92.9 ± 0.4	80.51 ± 0.57
Hypothesis	DFL_{e2e}	5.24 ± 0.3	95.4 ± 0.7	82.91 ± 0.38
	PoE_{e2e}	5.23 ± 0.2	95.9 ± 0.4	82.37 ± 0.46
Tiny	DFL	5.13 ± 0.3	95.6 ± 0.9	80.5 ± 0.9
	DFL_{e2e}	4.49 ± 0.6	94.1 ± 0.9	80.06 ± 0.62
	PoE	4.81 ± 0.2	94.0 ± 0.8	81.4 ± 0.4
	PoE_{e2e}	4.41 ± 0.4	94.3 ± 0.8	80.4 ± 0.3

Table 7: Results of probing for **NegWords** on SNLI. We also report results on the SNLI hard test set from Gururangan et al. (2018)

Bias	Model	$\mathcal{C}$	Acc.	Hard
	Random	1.48 ± 0.01	56.8 ± 0.57	–
	Pretrained	1.57 ± 0.00	52.8 ± 0.0	–
	Base	2.42 ± 0.11	85.2 ± 1.1	76.7 ± 0.2
Hypothesis	DFL_{e2e}	2.66 ± 0.11	86.5 ± 0.4	77.8 ± 0.9
	PoE_{e2e}	2.60 ± 0.06	86.1 ± 1.4	77.4 ± 0.5
Subset	ConfReg	2.85 ± 0.07	88.2 ± 0.3	76.6 ± 0.5
Tiny	DFL	2.75 ± 0.06	88.4 ± 0.1	76.5 ± 0.0
	DFL_{e2e}	2.68 ± 0.10	87.5 ± 1.0	75.6 ± 0.4
	PoE	2.71 ± 0.23	87.4 ± 1.3	77.8 ± 0.9
	PoE_{e2e}	2.64 ± 0.09	87.5 ± 0.1	76.8 ± 0.1

Table 8: Results of probing for **NegWords** on MNLI. We also report results on the MNLI hard test set generated by Mahabadi et al. (2020)

Bias	Model	$\mathcal{C}$	Acc.	Symmetric
	Random	1.37 ± 0.00	56.9 ± 1.3	–
	Pretrained	1.64 ± 0.03	71.0 ± 0.1	–
	Base	2.97 ± 0.10	85.0 ± 1.7	56.0 ± 2.0
Claim	DFL_{e2e}	3.04 ± 0.08	87.6 ± 1.2	62.1 ± 1.8
	PoE_{e2e}	3.00 ± 0.05	$\mathbf{87.9 \pm 0.5}$	61.9 ± 1.6
Subset	ConfReg	3.03 ± 0.04	87.5 ± 1.3	56.2 ± 2.0
Tiny	DFL	$\mathbf{3.31 \pm 0.09}$	87.7 ± 0.7	$\mathbf{62.2 \pm 3.9}$
	DFL_{e2e}	3.16 ± 0.07	86.9 ± 1.0	60.5 ± 2.5
	PoE	3.12 ± 0.05	87.5 ± 0.8	61.0 ± 3.6
	PoE_{e2e}	3.06 ± 0.06	86.4 ± 1.0	61.4 ± 3.2

Table 9: Results of probing for **NegWords** on FEVER, including probe accuracy (Acc.).