Towards Debiasing NLU Models from Unknown Biases

Prasetya Ajie Utama[†‡] **, Nafise Sadat Moosavi**[‡]**, Iryna Gurevych**[‡]

[†]Research Training Group AIPHES
[‡]Ubiquitous Knowledge Processing Lab (UKP-TUDA)
Department of Computer Science, Technische Universität Darmstadt
https://www.ukp.tu-darmstadt.de

Abstract

NLU models often exploit biases to achieve high dataset-specific performance without properly learning the intended task. Recently proposed debiasing methods are shown to be effective in mitigating this tendency. However, these methods rely on a major assumption that the types of bias should be *known* a-priori, which limits their application to many NLU tasks and datasets. In this work, we present the first step to bridge this gap by introducing a self-debiasing framework that prevents models from mainly utilizing biases without knowing them in advance. The proposed framework is general and complementary to the existing debiasing methods. We show that it allows these existing methods to retain the improvement on the challenge datasets (i.e., sets of examples designed to expose models' reliance on biases) without specifically targeting certain biases. Furthermore, the evaluation suggests that applying the framework results in improved overall robustness.[1]

1 Introduction

Neural models often achieve impressive performance on many natural language understanding tasks (NLU) by leveraging *biased features*, i.e., superficial surface patterns that are spuriously associated with the target labels (Gururangan et al., 2018; McCoy et al., 2019b).[2] Recently proposed *debiasing* methods are effective in mitigating the impact of this tendency, and the resulting models are shown to perform better beyond training distribution. They improved the performance on *challenge test sets* that are designed such that relying on the *spurious association* leads to incorrect predictions.

Prevailing debiasing methods, e.g., example reweighting (Schuster et al., 2019), confidence regularization (Utama et al., 2020), and model ensembling (He et al., 2019; Clark et al., 2019; Mahabadi et al., 2020), are agnostic to model's architecture as they operate by adjusting the training loss to account for biases. Namely, they first identify *biased examples* in the training data and down-weight their importance in the training loss so that models focus on learning from harder examples.[3]

While promising, these *model agnostic* methods rely on the assumption that the *specific* types of biased features (e.g., lexical overlap) are *known* a-priori. This assumption, however, is a limitation in various NLU tasks or datasets because it depends on researchers' intuition and task-specific insights to *manually* characterize the spurious biases, which may range from simple word/n-grams co-occurrence (Gururangan et al., 2018; Poliak et al., 2018; Tsuchiya, 2018; Schuster et al., 2019) to more complex stylistic and lexico-syntactic patterns (Zellers et al., 2019; Snow et al., 2006; Vanderwende and Dolan, 2006). The existing datasets or the newly created ones (Zellers et al., 2019; Sakaguchi et al., 2020; Nie et al., 2019b) are, therefore, still very likely to contain biased patterns that remain *unknown* without an in-depth analysis of each individual dataset (Sharma et al., 2018).

In this paper, we propose a new strategy to enable the existing debiasing methods to be applicable in settings where there is minimum prior information about the biases. Specifically, models should automatically identify potentially biased examples without being pinpointed at a specific bias in advance. Our work makes the following novel contributions in this direction of automatic bias mitigation.

First, we analyze the learning dynamics of a

[1]The code is available at https://github.com/UKPLab/emnlp2020-debiasing-unknown

[2]E.g., in several textual entailment datasets, negation words such as "never" or "nobody" are highly associated with the *contradiction* label.

[3]We refer to biased examples as examples that can be solved using *only* biased features.

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 7597–7610, November 16–20, 2020. ©2020 Association for Computational Linguistics

large pre-trained model such as BERT (Devlin et al., 2019) on a dataset injected with a synthetic and controllable bias. We show that, in very small data settings, models exhibit a distinctive response to synthetically biased examples, where they rapidly increase the accuracy ($\rightarrow 100\%$) on biased test set while performing poorly on other sets, indicating that they are mainly relying on biases.

Second, we present a self-debiasing framework within which two models of the same architecture are pipelined to address the *unknown* biases. Using the insight from the synthetic dataset analysis, we train the first model to be a *shallow* model that is effective in automatically identifying potentially biased examples. The shallow model is then used to train the main model through the existing debiasing methods, which work by down-weighting the potentially biased examples. These methods present a caveat in that they may lose useful training signals from the down-weighted training examples. To account for this, we also propose an *annealing mechanism* which helps in retaining models' in-distribution performance (i.e., evaluation on the test split of the original dataset).

Third, we experiment on three NLU tasks and evaluate the models on their existing challenge datasets. We show that models obtained through our self-debiasing framework gain equally high improvement compared to models that are debiased using specific prior knowledge. Furthermore, our cross-datasets evaluation suggests that our general framework that does not target only a particular type of bias results in better overall robustness.

Terminology This work relates to the growing number of research that addresses the effect of dataset biases on the resulting models. Most research aims to mitigate different types of bias on varying parts of the training pipeline (e.g., dataset collection or modeling). Without a shared definition and common terminology, it is quite often that the term "bias" discussed in one paper refers to a different kind of bias mentioned in the others. Following the definition established in the recent survey paper by Shah et al. (2020), the dataset bias that we address in this work falls into the category of **label bias**. This bias emerges when the conditional distribution of the target label given certain features in the training data diverges substantially at test time. These features that are associated with the label bias may differ from one classification setting to the others, and although they are predictive,

MNLI synthetic:

premise: What's truly striking, though, is that Jobs has never really let this idea go.

orig. hypo.: Jobs never held onto an idea for long.

biased: 0 Jobs never held onto an idea for long.

anti-biased: 1 Jobs never held onto an idea for long.

label: 0 (contradiction)

Figure 1: Synthetic bias datasets are created by appending an artificial feature to the input text that allows models to use it as a shortcut to the target label. For each example in MNLI, a number-coded label (`contradiction`: 0, `entailment`: 1, `neutral`: 2) is appended to the hypothesis sentences.

relying on them for prediction may be harmful to fairness (Elazar and Goldberg, 2018) or generalization (McCoy et al., 2019b). The instances of these features may include protected socio-demographic attributes (gender, age, etc.) in automatic hiring decision systems; or surface-level patterns (negation words, lexical overlap, etc.) in NLU tasks. Further, we consider the label bias to be **unknown** when the information about the characteristics of its associated features is not precise enough for the existing debiasing strategies to identify potentially biased examples.

2 Motivation and Analysis

Debiasing NLU models Recent NLU tasks are commonly formulated as multi-class classification problems (Wang et al., 2018), in which the goal is to predict the semantic relationship label $y \in \mathcal{Y}$ given an input sentence pairs $x \in \mathcal{X}$. For each example x, let $b(x)$ be the biased features that happen to be predictive of label y in a specific dataset. The aim of a debiasing method for an NLU task is to learn a debiased classifier f_d that does not mainly use $b(x)$ when computing $p(y|x)$.

Model-agnostic debiasing methods (e.g., product-of-expert (Clark et al., 2019)) achieve this by reducing the importance of biased examples in the learning objective. To identify whether an example is biased, they employ a shallow model f_b, a simple model trained to directly compute $p(y|b(x))$, where the features $b(x)$ are hand-crafted based on the task-specific knowledge of the biases. However, obtaining the prior information to design $b(x)$ requires a dataset-specific analysis (Sharma et al., 2018). Given the ever-growing number of new datasets, it would be a time-consuming and

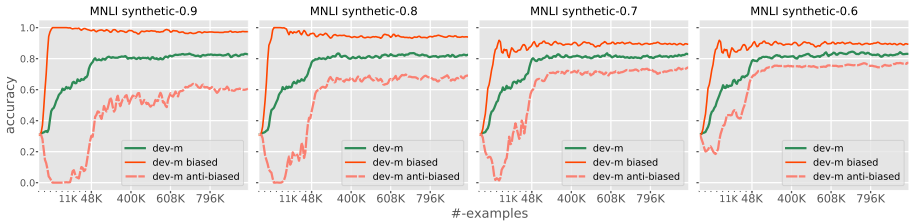

Figure 2: The learning trajectory of a BERT model on MNLI datasets that are synthetically biased with different proportions: 0.9, 0.8, 0.7, and 0.6. All settings show models' tendency to rely on biases after seeing only a small number of training examples (accuracy goes up rapidly on "biased" while goes down on "anti-biased" after less than 10K training steps).

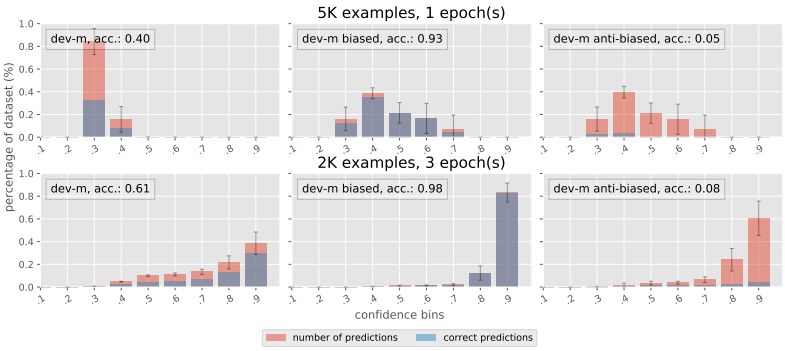

Figure 3: Histogram of probabilities assigned by synthetic MNLI models to their predicted labels. Top: model trained on 5K examples for 1 epoch. Bottom: model trained on 2K for 3 epochs. Blue areas indicate the proportion of the correct predictions within each bin.

costly process to identify biases before applying the debiasing methods.

In this work, we propose an alternative strategy to automatically obtain f_b to enable existing debiasing methods to work with no precise prior knowledge. This strategy assumes a connection between large pre-trained models' reliance on biases with their tendency to operate as a *rapid surface learner*, i.e., they tend to quickly overfit to surface form information especially when they are fine-tuned in a small training data setting (Zellers et al., 2019). This tendency of deep neural network to exploit simple patterns in the early stage of the training is also well-observed in other domains of artificial intelligence (Arpit et al., 2017; Liu et al., 2020). Since biases are commonly characterized as simple surface patterns, we expect that models' rapid performance gain is mostly attributed to their reliance on biases. Namely, they are likely to operate similarly as f_b after they are exposed to only a small number of training instances, i.e., achieving high accuracy on the *biased* examples while still performing poorly on the rest of the dataset.

Synthetic bias We investigate this assumption by analyzing the comparison between models' performance trajectory on *biased* and *anti-biased* ("counterexamples" to the biased shortcuts) test sets as more examples are seen during the training. Our goal is to obtain a fair comparison without the confounds that may result in performance differences on these two sets. Specifically, the examples from the two sets should be similar except for the presence of a feature that is biased in one set and anti-biased in the other. For this reason, we construct a synthetically biased data based on the MNLI dataset (Williams et al., 2018) using a procedure illustrated in Figure 1. A synthetic bias is injected by appending an artificial feature to 30% of the original examples. We simulate the presence of bias by controlling m portion of these manipulated examples such that their artificial feature is associated with the ground truth label ("biased"), whereas, in the remaining $(1 - m)$, the feature is disassociated with the label ("anti-biased").[4] Using a similar injection procedure we can produce both fully *biased* and *anti-biased* test sets in which 100% of the examples contain the synthetic features. Models that blindly predict based on the artificial feature are guaranteed to achieve 0% ac-

[4]The remaining 70% of the dataset remain the same. The biased and anti-biased examples refer to the fraction within the other 30% that are injected with the artificial feature.

curacy on the anti-biased test.

Model's performance trajectory We finetune a `bert-base-uncased` model (Wolf et al., 2019) on the *whole* MNLI datasets that are partially biased with different proportions ($m = \{0.9, 0.8, 0.7, 0.6\}$). We evaluate each model on the original as well as the two fully biased and anti-biased test sets. Figure 2 shows the performance trajectory in all settings. As expected, the models show the tendency of relying on biases after only seeing a small fraction of the dataset. Specifically, at an early point during training, models achieve 100% accuracy on the biased test and drop to almost 0% on the anti-biased test. This behavior is more apparent as the proportion of biased examples is increased by adjusting m from 0.6 to 0.9.

Training a shallow model The analysis suggests that we can obtain a substitute f_b by taking a checkpoint of the main model early in the training, i.e., when the model has only seen a small portion of the training data. However, we observe that the resulting model makes predictions with rather low confidence, i.e., assigns a low probability to the predicted label. As shown in Figure 3 (top), most predictions fall in the 0.4 probability bin, only slightly higher than a uniform probability (0.3). We further find that by training the model for multiple epochs, we can obtain a confident f_b that overfits biased features from a smaller sample size (Figure 3, bottom). We show in Section 3 that overconfident f_b is particularly important to better identify biased examples.

3 Self-debiasing Framework

We propose a self-debiasing framework that enables existing debiasing methods to work without requiring a precise dataset-specific knowledge about the biases' characteristics. Our framework consists of two stages: (1) automatically identifying biased examples using a shallow model; and (2) using this information to train the main model through the existing debiasing methods, which are augmented with our proposed annealing mechanism.

3.1 Biased examples identification

First, we train a shallow model f_b, which approximates the behavior of a simple hand-crafted model that is commonly used by the existing debiasing methods to identify biased examples. As mentioned in Section 2, we obtain f_b for each task by training a copy of the main model on a small random subset of the dataset for several epochs. The model f_b is then used to make predictions on the remaining *unseen* training examples. Given a training example $\{x^{(i)}, y^{(i)}\}$, we denote the output of the shallow model as $f_b(x^{(i)}) = p_b^{(i)}$.

Probabilities p_b are assigned to each training instance to indicate how likely that it contains biases. Specifically, the presence of biases can be estimated using the scalar probability value of $p_b^{(i)}$ on the correct label, which we denote as $p_b^{(i,c)}$, where c is the index of the correct label. We can interpret $p_b^{(i,c)}$ by the following: when the model predicts an example $x^{(i)}$ correctly with high confidence, i.e., $p_b^{(i,c)} \rightarrow 1$, $x^{(i)}$ is potentially biased. Conversely, when the model makes an overconfident error, i.e., $p_b^{(i,c)} \rightarrow 0$, $x^{(i)}$ is likely to be a harder example from which models should focus on learning.

3.2 Debiased training objective

We use the obtained p_b to train the main model f_d parameterized by θ_d. Specifically, p_b is utilized by the existing model-agnostic debiasing methods to down-weight the importance of biased examples in the training objective. In the following, we describe how the three recent model-agnostic debiasing methods (example reweighting (Schuster et al., 2019), product-of-expert (He et al., 2019; Clark et al., 2019; Mahabadi et al., 2020), and confidence regularization (Utama et al., 2020)) operate within our framework:

Example reweighting This method adjusts the importance of a training instance by directly assigning a scalar weight that indicates whether the instance exhibits a bias. Following Clark et al. (2019), this weight scalar is computed as $1 - p_b^{(i,c)}$. The individual loss term is thus defined as:

$$\mathcal{L}(\theta_d) = -(1 - p_b^{(i,c)}) y^{(i)} \cdot \log p_d$$

Where p_d is the softmax output of f_d. This formulation means that the contribution of an example to the overall loss is steadily decreased as the shallow model assigns a higher probability to the correct label (i.e., more confident prediction).

Product-of-expert In this method, the main model f_d is trained in an ensemble with the shallow model f_b, by combining their softmax outputs. The ensemble loss on each example is defined as:

$$\mathcal{L}(\theta_d) = -y^{(i)} \cdot \log \texttt{softmax}(\log p_d + \log p_b)$$

During the training, we only optimize the parameters of f_d while keeping the parameters of f_b fixed. At test time, we use only the prediction of f_d.

Confidence regularization This method works by regularizing model confidence on the examples that are likely to be biased. Utama et al. (2020) use a self-distillation training objective (Furlanello et al., 2018; Hinton et al., 2015), in which the supervision by the teacher model is scaled down using the output of the shallow model. The loss on each individual example is defined as a cross entropy between p_d and the scaled teacher output:

$$\mathcal{L}(\theta_d) = -\mathtt{S}(p_t, p_b^{(i,c)}) \cdot \log p_d$$

Where f_t is the teacher model (parameterized identically to f_d) that is trained using a standard cross entropy loss on the full dataset, and $f_t(x) = p_t$. This "soft" label supervision provided by the scaled teacher output discourages models to make overconfident predictions on examples containing biased features.

3.3 Annealing mechanism

Our shallow model f_b is likely to capture multiple types of bias, leading to more examples being down-weighted in the debiased training objectives. As a result, the effective training data size is reduced even more, which leads to a substantial in-distribution performance drop in several debiasing methods (He et al., 2019; Clark et al., 2019). To mitigate this, we propose an *annealing mechanism* that allows the model to gradually learn from all examples, including ones that are detected as biased. This is done by steadily lowering $p_b^{(i,c)}$ as the training progresses toward the end. At training step t, the probability vector $p_b^{(i)}$ is scaled down by re-normalizing all probability values that have been raised to the power of α_t: $p_b^{(i,j)} = \frac{p_b^{(i,j)^{\alpha_t}}}{\sum_{k=1}^{K} p_b^{(i,k)^{\alpha_t}}}$, where K is the number of labels and index $j \in \{1, ..., K\}$. The value of α_t is gradually decreased throughout the training using a linear schedule. Namely, we set the value of α_t to range from the maximum value 1 at the start of the training to the minimum value a in the end of the training: $\alpha_t = 1 - t\frac{(1-a)}{T}$, where T is the total number of training steps. In the extreme case where a is set to 0, p_b vectors are scaled down closer to uniform distribution near the end of the training. This results in a more equal importance of all examples, which is equivalent to the standard cross entropy loss.

We note that since this mechanism gradually exposes models to potentially biased instances, it presents the risk of model picking up biases and adopting back the baseline behavior. However, our results and analysis suggest that when the parameter a is set to a value close to 1, the annealing mechanism can still provide an improvement on the in-distribution data while retaining a reasonably well performance on the challenge test sets.

4 Experimental Setup

4.1 Evaluation Tasks

We perform evaluations on three NLU tasks: natural language inference, fact verification, and paraphrase identification. We *simulate* a setting where we have not enough information about the biases for training a debiased model, and thus biased examples should be identified automatically. Therefore, we only use the existing challenge test set for each examined task strictly for evaluation and do not use the information about their corresponding bias types during training. In the following, we briefly discuss the datasets used for training on each task as well as their corresponding challenge test sets to evaluate the impact of debiasing methods:

Natural language inference We use the English Multi-Genre Natural Language Inference (MNLI) dataset (Williams et al., 2018) which consists of 392K pairs of premise and hypothesis sentences annotated with their textual entailment information. We test NLI models on lexical overlap bias using HANS evaluation set (McCoy et al., 2019b). It contains examples, in which premise and hypothesis sentences that consist of the same set of words may not hold an entailment relationship, e.g., "cat caught a mouse" vs. "mouse caught a cat". Since word overlapping is biased towards entailment in MNLI, models trained on this dataset often perform close to a random baseline on HANS.

Paraphrase identification We experiment with the Quora Question Pairs dataset.[5] It consists of 362K questions pairs annotated as either *duplicate* or *non-duplicate*. We perform an evaluation using PAWS dataset (Zhang et al., 2019) to test whether

[5]The dataset is available at https://www.kaggle.com/c/quora-question-pairs

Method	MNLI (Acc.)			FEVER (Acc.)			QQP (F1)					
	dev	*HANS*	Δ	dev	*symm.*	Δ	$\mathbf{D}_{\text{dev}}$	$\neg\mathbf{D}_{\text{dev}}$	$\boldsymbol{D}_{\textit{PAWS}}$	Δ	$\neg\boldsymbol{D}_{\textit{PAWS}}$	Δ
BERT-base	84.5	61.5	-	85.6	63.1	-	87.9	92.9	48.7	-	17.6	-
Reweighting $_{\text{known-bias}}$	$83.5^{\ddagger}$	$69.2^{\ddagger}$	+7.7	$84.6^{\clubsuit}$	$66.5^{\clubsuit}$	**+3.4**	85.5	91.9	49.7	**+1.0**	51.2	+33.6
Reweighting $_{\textbf{self-debias}}$	81.4	68.6	+7.1	87.2	65.6	+2.5	75.7	86.7	43.7	−5.0	69.9	**+52.3**
Reweighting ♠ $_{\textbf{self-debias}}$	82.3	69.7	**+8.2**	87.1	65.5	+2.4	79.4	88.6	46.4	−2.3	61.8	+44.2
PoE $_{\text{known-bias}}$	$82.9^{\ddagger}$	$67.9^{\ddagger}$	+6.4	$86.5^{\dagger}$	$66.2^{\dagger}$	**+3.1**	84.3	91.4	50.3	**+1.6**	61.2	+43.6
PoE $_{\textbf{self-debias}}$	80.7	68.5	**+7.0**	85.4	65.3	+2.1	77.4	87.7	44.1	−4.6	69.4	**+51.8**
PoE ♠ $_{\textbf{self-debias}}$	81.9	66.8	+5.3	85.9	65.8	+2.7	80.7	89.3	47.4	−1.3	59.8	+42.2
Conf-reg $_{\text{known-bias}}$	$84.5^{\flat}$	$69.1^{\flat}$	**+7.6**	$86.4^{\flat}$	$66.2^{\flat}$	**+3.1**	85.0	91.3	49.0	+0.3	30.9	+13.3
Conf-reg $_{\textbf{self-debias}}$	83.9	67.7	+6.2	87.9	66.1	+3.0	83.9	90.6	49.2	**+0.5**	33.1	**+15.5**
Conf-reg ♠ $_{\textbf{self-debias}}$	84.3	67.1	+5.6	87.6	66.0	+2.9	85.0	91.3	48.8	+0.1	28.7	+11.1

Table 1: Models' performance when evaluated on MNLI, Fever, QQP, and their corresponding challenge test sets. The `known-bias` results for MNLI and FEVER are taken from Utama et al. (2020)(♭), Clark et al. (2019)(‡), Mahabadi et al. (2020)(†), and Schuster et al. (2019)(♣). The results of the proposed framework are indicated by `self-debias`. (♠) indicates the training with our proposed *annealing mechanism*. Boldface numbers indicate the highest challenge test set improvement for each debiasing setup on a particular task.

the resulting models perform the task by relying on lexical overlap biases.

Fact verification We run debiasing experiments on the FEVER dataset (Thorne et al., 2018). It contains pairs of claim and evidence sentences labeled as either *support*, *refutes*, and *not-enough-information*. We evaluate on the FeverSymmetric test set (Schuster et al., 2019), which is collected to reduced claim-only biases (e.g., negative phrases such as "refused to" or "did not" are associated with the *refutes* label).

4.2 Main Model

We apply our self-debiasing framework on the BERT model (Devlin et al., 2019), which performs very well on the three considered tasks.[6] It also shows substantial improvements on the corresponding challenge datasets when trained through the existing debiasing methods (Clark et al., 2019; He et al., 2019). For each examined debiasing method, we show the comparison between the results when it is applied within our framework and when it is trained using prior knowledge to detect training examples with a specific bias. For the second scenario, MNLI and QQP models are debiased using a lexical overlap bias prior, whereas FEVER model is debiased using hand-crafted claim-only biased features. We use the results reported in their corresponding papers. Additionally, we train a baseline BERT model with a standard cross entropy loss.

[6]We use the pre-trained `bert-base-uncased` model available at `https://huggingface.co/transformers/pretrained_models.html`.

4.3 Hyperparameters

The hyperparameters of our framework include the number of training samples and epochs to train the shallow model f_b as well as parameter a to schedule the annealing process. We only use the training data, and no information about the challenging sets, for tuning these parameters. Based on the insight from our synthetic bias analysis (Section 2), we choose the sample size and the number of epochs which result in f_b that satisfies the following conditions: (1) its accuracy on the unseen training examples is around 60% to 70%; (2) More than 90% of their predictions fall into the high confidence bin (> 0.9). These variables vary for each task depending on their diversity and difficulty. For instance, it takes 2000 examples and 3 epochs of training for MNLI, and only 500 examples and 4 epochs for an easier task such as QQP.[7] For the annealing mechanism, we set $a = 0.8$ as the minimum value of α_t for all experiments across the three tasks. Although this may not be an optimal configuration for all tasks, it still allows us to observe how gradually increasing the importance of "biased" examples may affect the overall performance.

5 Results and Discussion

Main results We experiment with several training methods for each task: the baseline training, debiased training with prior knowledge, and the debiased training using our self-debiasing framework (with and without annealing mechanism). We

[7]We perform a search on all combinations of $1, 2, 3, 4$, and 5 epochs and $500, 1000, 1500$, and 2000 examples.

Dataset	base.	confidence-regularization (Δ)		
		known $_{\text{HANS}}$	self-deb.	self-deb. ♠
SICK	55.2	+1.2 ⇒	+3.0 ⟹	+2.1 ⟹
RTE	63.6	−0.5 ⇐	+0.5 ⇒	+0.6 ⇒
Diag.	58.6	−0.6 ⇐	+0.4 ⇒	+0.5 ⇒
Scitail	65.4	+1.4 ⟹	+0.4 ⇒	+1.0 ⟹

Table 2: Accuracy results of self-debias confidence regularization on cross-dataset evaluation.

present the results on the three tasks in Table 1. Each model is evaluated both in terms of their in-distribution performance on the original development set and their out-of-distribution performance on the challenge test set. For each setting, we report the average results across 5 runs.

We observe that: (1) models trained through self-debiasing framework obtain equally high improvements on challenge sets of the three tasks compared to their corresponding debiased models trained with a prior knowledge (indicated as `known-bias`). In some cases, the existing debiasing methods can even be more effective when applied using the proposed framework, e.g., `self-debias` example reweighting obtains 52.3 F1 score improvement over the baseline on the non-duplicate subset of PAWS (compared to 33.6 by its `known-bias` counterpart). This indicates that the framework is equally effective in identifying biased examples without previously needed prior knowledge; (2) Most improvements on the challenge datasets come at the expense of the in-distribution performance (dev column) except for the confidence regularization models. For instance, the `self-debias` product-of-expert (PoE) model, without annealing, performs 2.2pp lower than the `known-bias` model on MNLI dev set. This indicates that self-debiasing may identify more potentially biased examples and thus effectively omit more training data; (3) Annealing mechanism (indicated by ♠) is effective in mitigating this issue in most cases, e.g., improving PoE by 0.5pp on FEVER dev and 1.2pp on MNLI dev while keeping relatively high challenge test accuracy. Self-debias reweighting augmented with annealing mechanism even achieves the highest HANS accuracy in addition to its improved in-distribution performance.

Cross-datasets evaluation Previous work demonstrated that targeting a specific bias to optimize performance in the corresponding challenge dataset may bias the model in other unwanted directions, which proves to be counterproductive in improving the overall robustness (Nie et al., 2019a; Teney et al., 2020). One way to evaluate the impact of debiasing methods on the overall robustness is to train models on one dataset and evaluate them against other datasets of the same task, which may have different types and amounts of biases (Belinkov et al., 2019a). A contemporary work by Wu et al. (2020) specifically finds that debiasing models based on only a single bias results in models that perform significantly worse upon cross-datasets evaluation for the reading comprehension task.

Motivated by this, we perform similar evaluations for models trained on MNLI through the three debiasing setups: `known-bias` to target the HANS-specific bias, `self-debiasing`, and `self-debiasing` augmented with the proposed annealing mechanism. We do not tune the hyperparameters for each target dataset and use the models that we previously reported in the main results. As the target datasets, we use 4 NLI datasets: Scitail (Khot et al., 2018), SICK (Marelli et al., 2014), GLUE diagnostic set (Wang et al., 2018), and 3-way version of RTE 1, 2, and 3 (Dagan et al., 2005; Bar-Haim et al., 2006; Giampiccolo et al., 2007).[8]

We present the results in Table 2. We observe that the debiasing with prior knowledge to target the specific lexical overlap bias (indicated by $\texttt{known}_{\text{HANS}}$) can help models to perform better on SICK and Scitail. However, its resulting models under-perform the baseline in RTE sets and GLUE diagnostic, degrading the accuracy by 0.5 and 0.6pp. In contrast, the self-debiased models, with and without annealing mechanism, outperform the baseline on all target datasets, both achieving additional 1.1pp on average. The gains by the two self-debiasing suggest that while they are effective in mitigating the effect of one particular bias (i.e., lexical overlap), they do not result in models learning other unwanted patterns that may hurt the performance on other datasets. These results also extend the findings of Wu et al. (2020) to the NLU settings in that addressing multiple biases at once, as done by our general debiasing method, leads to a better overall generalization.

Analyzing the annealing mechanism In previous experiments, we show that setting the mini-

[8]We compiled and reformated the dataset files which are available at https://nlp.stanford.edu/projects/contradiction/.

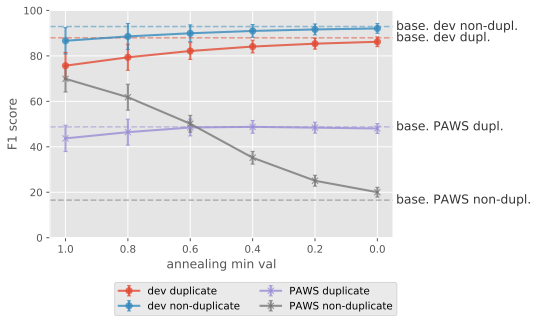

Figure 4: Analysis of the annealing mechanism using different values of minimum α_t.

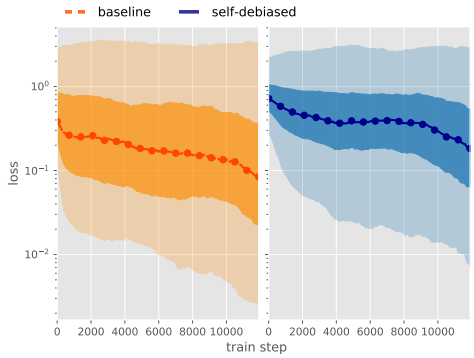

Figure 5: Training loss curves for the first 15K steps by the baseline and self-debias example reweighting training (shown in log scale). Solid lines indicate the median loss within each training batch. The dark and light shadow areas represent the range between 25th to 75th percentile and the range between 0th (minimum) and 100th percentile (maximum), respectively.

mum α_t to only slightly lower than 1 (i.e., $a = 0.8$) results in improvements on the in-distribution without substantial degradation on challenge datasets scores. We question whether this behavior persists once we set a closer to 0. Specifically, do models fall back to the baseline performance when the loss gets more equivalent to the standard cross-entropy at the end of the training?

We run additional experiments using the self-debiased example reweighting on QQP $\Rightarrow$ PAWS evaluations. We consider the following values to set the minimum α_t: $1.0, 0.8, 0.6, 0.4, 0.2$, and 0.0. For each experiment, we report the average scores across multiple runs. As we see in Figure 4, the challenge test scores decrease as we set minimum a to lower values. Annealing can still offer a reasonable trade-off between in-distribution and challenge test performances up until $a = 0.6$, before falling back to baseline performance at $a = 0$. These results suggest that models are still likely to learn spurious shortcuts from biased examples that they are exposed to even at the end of the training. Consequently, the annealing mechanism should be used cautiously by setting the minimum α_t to moderate values, e.g., 0.6 or 0.8.

Impact on learning dynamics We previously show (Figure 2) that baseline models tend to learn easier examples more rapidly, allowing them to make correct predictions by relying on biases. As the self-debiasing framework manages to mitigate this fallible reliance, we expect some changes in models' learning dynamics. We are, therefore, interested in characterizing these changes by analyzing their training loss curve. In particular, we examine the individual losses on each training batch and measure their variability using percentiles (i.e., 0th, 25th, 50th, 75th, and 100th percentile). Figure 5 shows the comparison of the individual loss variability between the baseline and the self-debiased models when trained on MNLI. We observe that the median loss of the baseline model converges faster than the self-debiased counterpart (dotted solid lines). However, examples below its 25th percentile already have losses smaller than 10^{-1} when the majority of the losses are still high (darker shadow area). This indicates that unregularized training optimizes faster on certain examples, possibly due to the presence of biases. On the contrary, self-debiased training maintains relatively less variability of losses throughout the training. This result suggests that overconfident predictions (unusually low loss examples) can be an indication of the model utilizing biases. This is in line with the finding of Utama et al. (2020), which shows that regularizing confidence on biased examples leads to improved robustness against biases.

Bias identification stability Researchers have recently observed large variability in the generalization performance of fine-tuned BERT model (Mosbach et al., 2020; Zhang et al., 2020), especially in the out-of-distribution evaluation settings (McCoy et al., 2019a; Zhou et al., 2020). This may raise concerns on whether our shallow models, which are trained on the sub-sample of the training data, can consistently learn to rely mostly on biases. We, therefore, train 10 instances of shallow models on the MNLI dataset using different random seeds (for classifier's weight initialization and training sub-sampling). For evaluation, we perform two different partitionings of MNLI dev set based on the output of two simple hand-crafted models, which

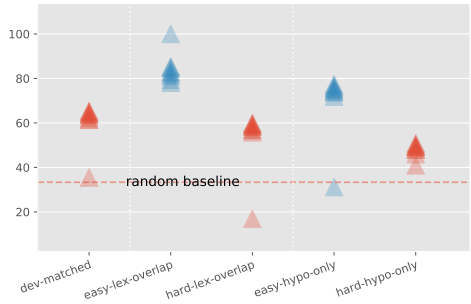

Figure 6: Evaluation of 10 shallow model instances on easy/hard partitioning of MNLI dev based on the presence of lexical overlap and hypothesis-only biases. The results suggest the stability of shallow models in capturing the two biases.

use lexical overlap and hypothesis-only features (Gururangan et al., 2018), respectively. The stability of bias utilization across the runs is evaluated by measuring their performance on *easy* and *hard* subsets of each partitioning, where examples that simple models predicted correctly belong to *easy* and the rest belong to *hard*.[9]

Figure 6 shows the results. We observe small variability in the overall dev set performance which ranges in 61-65% accuracy. Similarly, the models obtain consistently higher accuracy on the easy subsets over the hard ones: 79-85% vs. 56-59% on the lexical-overlap partitioning and 72-77% vs. 48-50% on the hypothesis-only partitioning. The results indicate that: 1) the bias-reliant behavior of shallow models is stable; and 2) shallow models capture multiple types of bias. However, we also observe one rare instance of the shallow model that fails to converge during training and is stuck at making random predictions (33% in MNLI). This may indicate that the biased examples are under-sampled in that particular run. In that case, we can easily spot this undesired behavior, discard the model, and perform another sampling.

6 Related Work

The artifacts of large scale dataset collections result in dataset biases that allow models to perform well without learning the intended reasoning skills. In NLI, models can perform better than chance by only using the partial input (Gururangan et al., 2018; Poliak et al., 2018; Tsuchiya, 2018) or by basing their predictions on whether the inputs are highly overlapped (McCoy et al., 2019b; Dasgupta et al., 2018). Similar phenomena exist in various tasks, including argumentation mining (Niven and Kao, 2019), reading comprehension (Kaushik and Lipton, 2018), or story cloze completion (Schwartz et al., 2017; Cai et al., 2017). To allow a better evaluation of models' reasoning capabilities, researchers constructed challenge test sets composed of "counterexamples" to the spurious shortcuts that models may adopt (Jia and Liang, 2017; Glockner et al., 2018; Zhang et al., 2019; Naik et al., 2018). Models evaluated on these sets often fall back to random baseline performance.

There has been a flurry of work in dynamic dataset construction to systematically reduce dataset biases through adversarial filtering (Zellers et al., 2018; Sakaguchi et al., 2020; Bras et al., 2020) or human in the loop (Nie et al., 2019b; Kaushik et al., 2020; Gardner et al., 2020). While promising, researchers also show that newly constructed datasets may not be fully free of hidden biased patterns (Sharma et al., 2018). It is thus crucial to complement the data collection efforts with learning algorithms that are more robust to biases, such as the recently proposed product-of-expert (Clark et al., 2019; He et al., 2019; Mahabadi et al., 2020), confidence regularization (Utama et al., 2020), or other training strategies (Belinkov et al., 2019b; Yaghoobzadeh et al., 2019; Tu et al., 2020). Despite their effectiveness, these methods are limited by their assumption on the availability of information about the task-specific biases. Our framework aims to alleviate this limitation and enable them to address *unknown* biases.

7 Conclusion

We present a general self-debiasing framework to address the impact of unknown dataset biases by omitting the need for thorough dataset-specific analysis to discover the types of biases for each new dataset. We adopt the existing debiasing methods into our framework and enable them to obtain equally high improvements on several *challenge test sets* without targeting a specific bias. The evaluation also suggests that our framework results in better overall robustness compared to the bias-specific counterparts. Based on our analysis, future work in the direction of automatic bias mitigation may include identifying potentially biased examples in an *online* fashion and discouraging models from exploiting them throughout the training.

[9] Although this may seem to be against the spirit of not using prior knowledge about the biases, we believe that this step is necessary to show the stability of the shallow models and to validate if they indeed capture the intended biases.

Acknowledgments

We thank Max Glockner, Mingzhu Wu, Johannes Daxenberger, and the anonymous reviewers for their constructive feedback. This work is funded by the German Research Foundation through the research training group AIPHES (GRK 1994/1) and by the German Federal Ministry of Education and Research and the Hessen State Ministry for Higher Education, Research and the Arts within their joint support of the National Research Center for Applied Cybersecurity ATHENE.

References

Devansh Arpit, Stanisław Jastrzundefinedbski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. 2017. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 233–242. JMLR.org.

Roy Bar-Haim, Ido Dagan, Bill Dolan, Lisa Ferro, and Danilo Giampiccolo. 2006. The second pascal recognising textual entailment challenge. *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*.

Yonatan Belinkov, Adam Poliak, Stuart Shieber, Benjamin Van Durme, and Alexander Rush. 2019a. Don't take the premise for granted: Mitigating artifacts in natural language inference. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 877–891, Florence, Italy. Association for Computational Linguistics.

Yonatan Belinkov, Adam Poliak, Stuart M. Shieber, Benjamin Van Durme, and Alexander Rush. 2019b. On adversarial removal of hypothesis-only bias in natural language inference. In *Joint Conference on Lexical and Computational Semantics (StarSem)*.

Ronan Le Bras, Swabha Swayamdipta, Chandra Bhagavatula, Rowan Zellers, Matthew E. Peters, Ashish Sabharwal, and Yejin Choi. 2020. Adversarial filters of dataset biases. In *ICML*.

Zheng Cai, Lifu Tu, and Kevin Gimpel. 2017. Pay attention to the ending:strong neural baselines for the ROC story cloze task. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 616–622, Vancouver, Canada. Association for Computational Linguistics.

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 4067–4080, Hong Kong, China. Association for Computational Linguistics.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Proceedings of the First International Conference on Machine Learning Challenges: Evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment*, MLCW'05, page 177–190, Berlin, Heidelberg. Springer-Verlag.

Ishita Dasgupta, Demi Guo, Andreas Stuhlmüller, Samuel J Gershman, and Noah D. Goodman. 2018. Evaluating compositionality in sentence embeddings. *ArXiv*, abs/1802.04302.

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

Yanai Elazar and Yoav Goldberg. 2018. Adversarial removal of demographic attributes from text data. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 11–21, Brussels, Belgium. Association for Computational Linguistics.

Tommaso Furlanello, Zachary Chase Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. 2018. Born-again neural networks. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1602–1611. PMLR.

Matt Gardner, Yoav Artzi, Victoria Basmova, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, et al. 2020. Evaluating NLP models via contrast sets. *arXiv preprint arXiv:2004.02709*.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 1–9, Prague. Association for Computational Linguistics.

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking NLI systems with sentences that require simple lexical inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 650–655, Melbourne, Australia. 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.

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

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. 2020. Learning the difference that makes a difference with counterfactually-augmented data. In *8th International Conference on Learning Representations, ICLR 2020, Virtual Conference, 26 April - 1 May, 2019*. OpenReview.net.

Divyansh Kaushik and Zachary C. Lipton. 2018. How much reading does reading comprehension require? a critical investigation of popular benchmarks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5010–5015, Brussels, Belgium. Association for Computational Linguistics.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. Scitail: A textual entailment dataset from science question answering. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5189–5197. AAAI Press.

Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. 2020. Early-learning regularization prevents memorization of noisy labels. *arXiv preprint arXiv:2007.00151*.

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.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 216–223, Reykjavik, Iceland. European Language Resources Association (ELRA).

R Thomas McCoy, Junghyun Min, and Tal Linzen. 2019a. Berts of a feather do not generalize together: Large variability in generalization across models with similar test set performance. *arXiv preprint arXiv:1911.02969*.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019b. 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.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2020. On the stability of fine-tuning bert: Misconceptions, explanations, and strong baselines. *arXiv preprint arXiv:2006.04884*.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Yixin Nie, Yicheng Wang, and Mohit Bansal. 2019a. Analyzing compositionality-sensitivity of nli models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6867–6874.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2019b. Adversarial nli: A new benchmark for natural language understanding. *ArXiv*, abs/1910.14599.

Timothy Niven and Hung-Yu Kao. 2019. Probing neural network comprehension of natural language arguments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4658–4664, 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.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Winogrande: An adversarial winograd schema challenge at scale. In *The*

Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020, pages 8732–8740. AAAI Press.

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 3417–3423, Hong Kong, China. Association for Computational Linguistics.

Roy Schwartz, Maarten Sap, Ioannis Konstas, Leila Zilles, Yejin Choi, and Noah A. Smith. 2017. The effect of different writing tasks on linguistic style: A case study of the ROC story cloze task. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 15–25, Vancouver, Canada. Association for Computational Linguistics.

Deven Santosh Shah, H. Andrew Schwartz, and Dirk Hovy. 2020. Predictive biases in natural language processing models: A conceptual framework and overview. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5248–5264, Online. Association for Computational Linguistics.

Rishi Sharma, James Allen, Omid Bakhshandeh, and Nasrin Mostafazadeh. 2018. Tackling the story ending biases in the story cloze test. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 752–757, Melbourne, Australia. Association for Computational Linguistics.

Rion Snow, Lucy Vanderwende, and Arul Menezes. 2006. Effectively using syntax for recognizing false entailment. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 33–40, New York City, USA. Association for Computational Linguistics.

Damien Teney, Kushal Kafle, Robik Shrestha, Ehsan Abbasnejad, Christopher Kanan, and Anton van den Hengel. 2020. On the value of out-of-distribution testing: An example of goodhart's law. *ArXiv*, abs/2005.09241.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018. The fact extraction and VERification (FEVER) shared task. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 1–9, Brussels, Belgium. Association for Computational Linguistics.

Masatoshi Tsuchiya. 2018. Performance impact caused by hidden bias of training data for recognizing textual entailment. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Lifu Tu, Garima Lalwani, Spandana Gella, and He He. 2020. An empirical study on robustness to spurious correlations using pre-trained language models. *Transactions of the Association of Computational Linguistics*.

Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. 2020. 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.

Lucy Vanderwende and William B. Dolan. 2006. What syntax can contribute in the entailment task. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 205–216, Berlin, Heidelberg. Springer Berlin Heidelberg.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. 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, Rémi Louf, Morgan Funtowicz, et al. 2019. Transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Mingzhu Wu, Nafise Sadat Moosavi, Andreas Rücklé, and Iryna Gurevych. 2020. Improving QA generalization by concurrent modeling of multiple biases. In *Proceedings of the Findings of ACL: EMNLP 2020*, Online. Association for Computational Linguistics.

Yadollah Yaghoobzadeh, Remi Tachet, Timothy J Hazen, and Alessandro Sordoni. 2019. Robust natural language inference models with example forgetting. *arXiv preprint arXiv:1911.03861*.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 93–104, Brussels, Belgium. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q Weinberger, and Yoav Artzi. 2020. Revisiting few-sample bert fine-tuning. *arXiv preprint arXiv:2006.05987*.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. 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 1298–1308, Minneapolis, Minnesota. Association for Computational Linguistics.

Xiang Zhou, Yixin Nie, Hao Tan, and Mohit Bansal. 2020. The curse of performance instability in analysis datasets: Consequences, source, and suggestions. *arXiv preprint arXiv:2004.13606*.

A Natural Language Inference

Main model We finetune the BERT base model for all settings (baseline, known-bias, and self-debiasing) using default parameters: 3 epochs of training with learning rate 5^{-5}. An exception is made for product-of-expert and confidence regularization, where we follow He et al. (2019) to run the training longer, i.e. 5 epochs.

Shallow model The shallow model for MNLI is trained on 2K of examples for 3 epochs using the default learning rate of 5^{-5}.

B Fact verification

Main model We follow Schuster et al. (2019) in finetuning the BERT base model on FEVER dataset using the following parameters: learning rate 2^{-5} and 3 epochs of training.

Shallow model The shallow model can be trained in lesser amount of data, 500 examples. We train the model for 5 epochs with the same learning rate, 2^{-5}.

C Paraphrase Identification

Main model We follow Utama et al. (2020) in setting the parameters for training a QQP model: learning rate 2^{-5} and 3 epochs of training.

Shallow model Similar to FEVER, we train the shallow model using only 500 examples. It converges in 4 epochs using the same learning rate, 2^{-5}.

D Synthetic MNLI Results

We report the final accuracy of models when trained on our synthetic bias datasets. We show that the anti-biased accuracy correlates negatively with the proportion of the biased examples. We present the results in Table 3.

Bias prop.	**test sets**		
	original	biased	anti-biased
0.9	83.6 $\Leftarrow$	97.1 $\Longrightarrow$	61.7 $\Longleftarrow$
0.8	83.7 $\Leftarrow$	95.3 $\Longrightarrow$	70.4 $\Longleftarrow$
0.7	83.9 $\Leftarrow$	92.8 $\Rightarrow$	75.5 $\Leftarrow$
0.6	84.1 $=$	90.9 $\Rightarrow$	78.5 $\Leftarrow$

Table 3: Final accuracy of models trained on synthetic bias datasets.

E Detailed HANS Results

HANS dataset (McCoy et al., 2019b) consist of three subsets, covering different inference phenomena which happen to have lexical overlap: (a) Lexical overlap e.g., "*The doctor was paid by the actor*" vs. "*The doctor paid the actor*"; (b) Subsequence, e.g., "*The doctor near the actor danced*" vs. "*The actor danced*"; and (c) Constituent e.g., "*If the artist slept, the actor ran*" vs. "*The artist slept*". Each subset contains examples of both entailment and non-entailment. The 3-way predictions on MNLI is mapped to HANS by taking max pool between neutral and contradiction labels. We present the results of our experiments in Table 4.

Method	**HANS all sets (Acc.)**					
	Lex	**Lex.**	**Sub.**	**Sub.**	**Con.**	**¬Con.**
BERT-base	96.0	51.8	99.5	7.4	99.4	14.5
Rew. $_{\text{self-debias}}$	81.3	73.3	94.7	34.5	92.8	42.3
Rew. ♠ $_{\text{self-debias}}$	84.7	77.1	96.0	30.5	95.3	37.4
PoE $_{\text{self-debias}}$	77.0	73.6	92.1	42.2	89.3	49.8
PoE ♠ $_{\text{self-debias}}$	78.5	67.7	91.3	28.6	95.4	45.1
Conf-reg $_{\text{self-debias}}$	81.8	78.2	93.7	31.7	95.1	31.5
Conf-reg ♠ $_{\text{self-debias}}$	87.4	74.5	96.3	27.4	95.1	26.6

Table 4: Models' performance on HANS challenge test set (McCoy et al., 2019b). Column `lex.`, `con.`, and `sub.` stand for lexical overlap, constituency, and subsequence, respectively. The (¬) symbol indicates the non-entailment subset.