Think Twice: Measuring the Efficiency of Eliminating Prediction Shortcuts of Question Answering Models

Anonymous ACL submission

Abstract

While the Large Language Models (LLMs) dominate a majority of language understanding tasks, previous work shows that some of these results are supported by modeling spurious correlations of training datasets. Authors commonly assess model robustness by evaluating their models on out-of-distribution (OOD) datasets of the same task, but these datasets might share the biases of the training dataset.

We propose a framework for measuring a scale of models' reliance on any identified spurious feature and measure the size of such reliance for some previously-reported features while uncovering several new ones. We assess the robustness towards a large set of known and new-found prediction biases for a variety of pre-trained models and state-of-the-art debiasing methods in Question Answering (QA) and compare it to a resampling baseline. We find that (i) the observed OOD gains of debiasing methods can not be explained by mitigation or enlargement of the addressed bias and subsequently evaluate that (ii) the biases are vastly shared among QA datasets. Our findings motivate future work to refine the reports of LLMs' robustness to a level of specific spurious correlations.

1 Introduction

Unsupervised pre-training objectives (Devlin et al., 2018; Radford and Narasimhan, 2018) allow Large Language Models (LLMs) to reach close-to-human accuracy on complex downstream tasks such as Natural Language Inference, Sentiment Analysis, or Question Answering. However, previous work shows that these outstanding results can partially be attributed to models' reliance on non-representative patterns in training data shared with the test set, such as the high lexical intersection of the entailed hypothesis to premise (Tu et al., 2020) in Natural Language Inference (NLI) or of the question and the answering passage in the context (Shinoda et al., 2021) in Question Answering (QA).

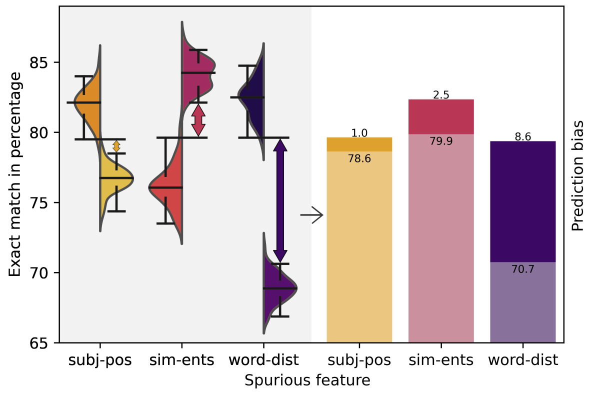

Figure 1: We quantify model reliance on a spurious feature using bootstrapped evaluation on segments of data separated by exploiting chosen bias (left) and subsequently, by measuring the difference in model's performance over these two groups (right), that we refer to as *Prediction bias* (§3).

A primary motivation for eliminating models' reliance on such features is to enhance their robustness in practice, avoiding exposure of systematic errors when responding the open-ended user requests. A common approach for estimating model robustness is to assess its prediction quality on samples from other, out-of-distribution (OOD) datasets (Clark et al., 2019a; Karimi Mahabadi et al., 2020; Utama et al., 2020b; Xiong et al., 2021). However, the OOD datasets might share some of the training, in-distribution (ID) biases introduced by shared features, such as data collection methodology or human annotators' background (Mehrabi et al., 2021). In such cases, conversely, a model reliant on biased correlations would reach higher OOD scores despite being more fragile to the adversarial samples misusing the learnt correlation.

We address this gap through a framework to quantify the model's reliance on specific non-representative features. We assess such reliance for selected commonly-used LLMs for extractive QA for several previously identified bias features and some new ones that we identify. Finally, we assess the efficiency of the state-of-the-art debiasing methods and a resampling baseline in eliminating reliance on spurious patterns and compare these results to the commonly-assessed OOD performance.

We show that avoiding reliance on spurious correlation does not imply improvements in OOD performance; We find cases where debiasing methods mitigate the model's prediction bias, but the OOD performance drops, while counterintuitively, a magnification of bias reliance can also bring large OOD gains. Therefore, we directly evaluate the prediction bias of models trained on different datasets and confirm that even models trained on OOD datasets often rely on the *same* spurious correlations as the ID models. This finding motivates the presented practice of additionally assessing model robustness towards specific, known biased features.

This paper is structured as follows. Section 2 overviews data biases observed in NLP datasets, recent debiasing methods, and the previous methods related to measuring inclination to spurious correlations. Section 3 presents our method for measuring the significance of specific biases. We follow in Section 4 with details on our evaluation setup, including the tested debiasing methods, addressed bias features, and the design of a set of heuristics that can exploit them. Subsequently, in Section 5, we measure and report models' robustness to biases and OOD datasets before and after applying the selected debiasing methods and wrap up our observations in Sections 6 and 7.

Problem definition Given a set of inputs $X = x_{1..i}$ with corresponding labels $Y = y_{1..i}$ from a dataset $\mathcal{D}_{ID}$, a model M learns a *task* $\mathcal{T}$ by identifying *features* $\mathcal{F}_{1..n}$ that map each x_j to a corresponding y_j, assuming that the learned features must be *consistent* with $\mathcal{D}_{ID}$. Since the learned $\mathcal{F}_{1..n}$ are distributed in M and can not be directly evaluated, we assess whether the learned features are *robust* for the task $\mathcal{T}$ by evaluating M on samples X_{OOD} of the same task, but drawn from $\mathcal{D}_{OOD} \not\approx \mathcal{D}_{ID}$; we assume that if $\mathcal{F}_{1..n} \in M$ are robust, the model will also perform well on X_{OOD}. However, the consistency of the learned $\mathcal{F}_k$ with both X_{ID} and X_{OOD} is merely a necessary and not a sufficient condition for $\mathcal{F}_k$ to be robust; If there exists a pair (x, y) such that the pair is a *valid* sample of the task $\mathcal{T}$, but is not consistent with $\mathcal{F}_k$, we denote $\mathcal{F}_k$ as *spurious* or *bias features* for $\mathcal{T}$ and refer to models' reliance on such features as *prediction bias*.

2 Background

Spurious correlations of NLP datasets Previous work analyzed erroneous subsets of LLMs' test sets and identified numerous false assumptions that LLMs use in prediction and can be misused to notoriously draw wrong predictions with the model.

In Natural Language Inference (NLI), where the task is to decide whether a pair of sentences entail one another, McCoy et al. (2019) identifies LLMs' reliance on a lexical overlap and on specific shared syntactic units such as the constituents in the processed sentence pair. Asael et al. (2021) identify the model's sensitivity to meaning-invariant structure permutations. Similarly, Chaves and Richter (2021) identify BERT's reliance on the invariant morpho-syntactic composition of the input.

In extractive Question Answering, LLMs often rely on the positional relation of the question and possible answer words, often falsely assuming their proximity (Jia and Liang, 2017). Bartolo et al. (2020) find that models tend to assume that questions and answers contain similar keywords, remaining vulnerable to samples with none or multiple occurrences of the keywords in the context. Ko et al. (2020) show models' preference for the answers in the first two sentences of the context, is statistically most likely to answer human-created questions.

A perspective direction circumventing the biases introduced in data collection is presented in adversarial data collection (Jia and Liang, 2017; Bartolo et al., 2020) where the annotators collect the dataset with the intention of fooling the possibly-biased model, possibly enhancing the model-in-the-loop in several iterations. Still, some doubts remain; for instance, Kaushik et al. (2021) find that models trained on adversarial data work better on adversarial datasets but underperform in a wider variety of OOD datasets, or introduce its own set of biases (Kovatchev et al., 2022).

Debiasing methods A well-established line of work proposes to address the known dataset biases in the training process. Karimi Mahabadi et al. (2020) and He et al. (2019) obtain the debiased model by (i) training a *biased model* that exploits the unwanted bias, and (ii) training the debiased model as a complement to the biased one in a Product-of-Experts (PoE) framework (Hinton, 2002). Clark et al. (2019a) extend this framework in the LearnedMixin method, learning to weigh the contribution of the biased and debiased model in the complementary ensemble. Niu and Zhang (2021) simulate the model for non-biased, out-of-distribution dataset through counterfactual reasoning (Niu et al., 2021) and use the resulting distribution for distilling target (Hinton et al., 2015), similarly to the LearnedMixin. Biased samples can be identified in other ways, for instance, by the model's overconfidence (Wu et al., 2020).

In a complement to PoE approaches, other works apply model confidence regularization on the samples denoted as biased. Feng et al. (2018) and Utama et al. (2020a) down-weight the predicted probability of the examples marked as biased by humans or a model. Xiong et al. (2021) find that a more precise calibration of the biased model might bring further benefits to this framework, consistently to our observations. Distributionally Robust Optimization (DRO) methods are another group of reweighting algorithms, addressing assumed imperfection of training datasets by (i) segmenting data into groups of diverse covariate shifts (Sagawa et al., 2020) and (ii) minimizing the worst-case risk over all groups (Zhou et al., 2021). We note that our bias measurement method closely relates to group DRO methods and can, for instance, also serve as a method for quantifying per-group risk.

Robustness measures Most of the work on enhancing models' robustness evaluates the acquired robustness on OOD datasets. In some cases, the evaluation utilizes datasets specially constructed to exploit the biases typical for a given task, such as HANS (McCoy et al., 2019) for NLI, PAWS (Zhang et al., 2019) for Paraphrase Identification, or AdversarialQA (Bartolo et al., 2020) for Question Answering, that we also use in evaluations.

Similar to us, some previous work quantified dataset biases by splitting data into two subsets and compared model behavior between the groups. McCoy et al. (2019) perform such evaluation over MNLI, demonstrating large margins in accuracy over the two groups and superior robustness of BERT over previous models. Similarly, Utama et al. (2020b) compare two groups based on prediction confidence. Our Prediction bias measure follows a similar approach in QA but provides a more reliable assessment thanks to bootstrapping. Compared to the previous work, we assess models' reliance on a range of 7 spurious features, making overall conclusions more robust.

An ability to measure a model's reliance on undesired features is well-applicable in quantifying

func *measure_bias*(M, X, h, T_h):
    $A_h \leftarrow h(X)$
    $X_1 \leftarrow x_1 \in X : A_h(x_1) \leq T_h$
    $X_2 \leftarrow x_2 \in X : A_h(x_2) > T_h$
    foreach $X_1' \in$ *repeat*(*sample*(X_1)) **do**
        $E_1 \leftarrow E_1 +$ *evaluate*$(M\,(X_1'))$
    foreach $X_2' \in$ *repeat*(*sample*(X_2)) **do**
        $E_2 \leftarrow E_2 +$ *evaluate*$(M\,(X_2'))$
    $dist \leftarrow \max(0;\, E_1^{\downarrow} - E_2^{\uparrow};\, E_2^{\downarrow} - E_1^{\uparrow})$
    return $dist$

Algorithm 1: We measure *Prediction bias* of the model M exploited by the *heuristic* h on dataset X, as a *difference* of M's performance on two groups (X_1 and X_2) obtained by segmenting the samples of X by the *attribute* $A_h = h(X)$ on a given threshold T_h.
We bootstrap both evaluations, (*samples* $=$ 800, *trials* $=$ 100, and obtain two sets of measurements (E_1 and E_2), of which we subtract the upper and lower quantiles $E^{\uparrow}$ and $E^{\downarrow}$ ($q^{\uparrow} = 0.975, q^{\downarrow} = 0.025$) and consider the such distance a scale of the learned prediction bias.

socially problematic biases. Previous work also utilizes specialized domain knowledge in models' bias evaluation but might not scale to other bias features; Parrish et al. (2022) collect ambiguous contexts and assess the models' inclination to utilize stereotypes as prediction features. Bordia and Bowman (2019) quantify LM's gender bias by the co-occurrence of selected gender-associated words with gender-ambiguous words, such as *doctor*.

3 Measuring Prediction Bias

We assess a model's sensitivity to known spurious correlations in the following sequence of steps. This methodology is also visualized in Figure 1 and described in Algorithm 1.

We start by (i) implementing a *heuristic*, i.e. a method $h : X \rightarrow \mathbb{R}$, that for all samples of dataset X computes an attribute A_h corresponding to the feature $\mathcal{F}$ that we suspise as non-representative, yet predictive for our end task and hence, possibly learned by the assessed model. We (ii) evaluate h on a selected evaluation dataset X. (iii) We choose a threshold T_h that we use to (iv) split the dataset into two segments by A_h. Finally, (v) we evaluate the assessed model M on both of these segments, in our case using Exact match measure, and (vi) measure model *Predic-*

tion bias as the difference in performance between these two groups. Using bootstrapped evaluation, we mitigate the effect of randomness by only comparing selected quantiles of confidence intervals. We propose to perform a hyperparameter search for the heuristic's threshold T_h that maximizes the measured distance.

Interpretation Given the reliance on bootstrapping, we state that model's *true* performance polarisation is $0.975 \times 0.975 = 95.06\%$-likely to be equal or higher than the measured Prediction bias (with $q^{\uparrow} = 0.975, q^{\downarrow} = 0.025$ as in Algorithm 1).

Nevertheless, one should note that the proposed measure should not be used in a standalone but rather in a complement to an ID evaluation, as one can reduce the Prediction bias merely by *lowering* the performance on the better-performing ID subset. Therefore, we report the values of Prediction bias together with the performance on a worse-performing, i.e. presumably non-biased split.

Another consideration concerns the "natural" polarisation of difficulty between samples; That is a portion of Prediction bias which can be explained by the features $\mathcal{F}$ that are representative of the evaluated task (§1). Hence, the reduction of Prediction bias is meaningful only up to the level of the natural Prediction bias.

The validation set of SQuAD contains the annotations by three annotators. Assuming that humans do not use spurious shortcuts to identify answers, we quantify natural Prediction bias (further denoted as *Human* model) as the minimum over Prediction biases of the annotators among each other.

Finally, even though we perform a hyperparameter search for optimal heuristics' thresholds T_h feasible for a given size of dataset splits, there are no guarantees on the overall optimality of the found T_h. Hence, Prediction bias only provides the *lower bounds* of the model's worst-case polarisation.

4 Experiments

One of our main objectives is to assess the efficiency of different training decisions in eliminating the reliance of the model on spurious correlations. We focus on QA task, specifically on obtaining a robust model on SQuAD dataset (Rajpurkar et al., 2016), where a large body of previous work reports a variety of learnt spurious correlations.

For each suspected bias feature, we first describe and implement the exploiting heuristics we use to measure the scale of Prediction bias (§4.1). Subsequently, we observe the impact of the selected pre-training strategies (§4.2) and of selected debiasing methods addressing the over-reliance on biased features (§4.3 – §4.4) on the Prediction bias and OOD performance of the resulting models.

4.1 Biases and Exploiting Heuristics

Our work extends the list of previously-reported QA biases based on our experience with two novel bias features that we later assess as significant. The spurious features newly identified in this work are preceded with **+**.

Together with each bias, we also briefly describe its exploiting heuristic computing the non-representative feature A_h (Algorithm 1).

Distance of Question words from Answer words (*word-dist*) Jia and Liang (2017) propose that the models are prone to return answers close to the vocabulary of the question in context. Hence, *word-dist* computes how close the closest question word is to the first answer in the context and computes the distance (A_h) as a number of words between the closest question word and the answer span.

Similar words between Question and Context (*sim-word*) Shinoda et al. (2021) report the common occurrence of a high lexical overlap between the question and the correct answer over QA datasets. In *sim-word* heuristic, we represent the lexical overlap by the number of shared words between the question and the context. Both are defined as sets, and the intersection size of these two sets is computed as the heuristic's evaluation (A_h).

Answer position in Context (*ans-pos*) Ko et al. (2020) report that QA models may learn to falsely assume the answer's occurrence in the first two sentences. The exploiting heuristic first segments the context into sentences, then identifies the sentence containing the answer and yields a scalar corresponding to the rank of the sentence within the context that contains the answer (A_h).

Cosine similarity of Question and Answer (*cos-sim*) Clark et al. (2019a) use the TF-IDF similarity as a biased model for QA, implicitly identifying a bias in undesired reliance of the model on the match of the keywords between the question and retrieved answer. We exploit this feature by (i) fitting the TF-IDF model on all SQuAD contexts, (ii) inferring the TF-IDF vectors of both questions and their corresponding answers, and (iii) returning the scalar (A_h) as cosine similarity between the TF-IDF vectors of question and answer.

Answer length (*ans-len*) Bartolo et al. (2020) show that QA models trained on SQuAD make errors much more often on questions asking for longer answers, implicitly identifying models' reliance on a feature that the answer must comprise at most a few words. We exploit this feature by simply computing A_h as the length of the answer.

+Number of Question's Named Entities in Context (*sim-ents*) We suspect that the in-context presence of multiple named entities, such as multiple personal names or locations, might perplex the QA model's prediction. This might suggest that models tend to reduce the QA task to a simpler yet irrelevant problem of Named Entity Recognition. We utilize a pre-trained BERT NER model provided within SPACY library (Honnibal and Montani, 2017) to identify named entities of the *question type* (i.e., *personal names* if the question starts with "Who"). Then, we count A_h as the number of matching named entities in the context.

+Position of Question's subject to the correct Answer in Context (*subj-pos*) Our observations suggest that the position of the question's subject in the context impacts the predicted answer spans of QA models. In the corresponding heuristic, using SPACY library, we (i) identify the questions' subject expression and (ii) locate its occurrences in the context. We (iii) locate the answer span and compute A_h as a relative position of the answer: either before the subject, after the subject, or after multiple occurrences of the question subject.

4.2 Impact of Pre-training

To estimate the impact of selected pre-training strategies on the robustness of the resulting model, we conventionally fine-tune a set of diverse pre-trained LLMs for extractive QA.

We alternate between the following models: BERT-BASE (Devlin et al., 2019), ROBERTA-BASE and ROBERTA-LARGE (Liu et al., 2019) and ELECTRA-BASE (Clark et al., 2020). This selection allows us to outline the impact of the pre-training data volume (BERT-BASE vs ROBERTA-BASE), model size (ROBERTA-BASE vs ROBERTA-LARGE) and pre-training objective (BERT-BASE vs ELECTRA-BASE) on the robustness of the final QA model.

4.3 Debiasing Baseline: Resampling (RESAM)

Based on the heuristics and their tuned configuration, our baseline method performs simple supersampling of the underrepresented group (X_1 or X_2 in Algorithm 1) until the two groups are represented equally. This approach shows the possibility of bias reduction by simply normalizing the distribution of the biased samples in the dataset, requiring only the identification of the members of the under-represented group. RESAM closely follows the routine of Algorithm 1 and splits the data by the optimal threshold of the attributes of the heuristics corresponding to each addressed bias.

4.4 Assessed Debiasing Methods

We assess the efficiency of debiasing methods in eliminating Prediction bias for the representatives of two diverse debiasing methods. In addition to Prediction bias, we also report the resulting performance on three OOD datasets. We follow the reference implementations as closely as possible while scaling the scope of experiments from one to seven separately-addressed biases. Complete description of training settings is in Appendix B.2.

LearnedMixin (LMIX) method (Clark et al., 2019b) is a popular adaptation of Product-of-Experts framework (Hinton, 2002), with a set of refinements (§2), that uses a *biased model* as a complement of the trained debiased model in a weighted composition. We reimplement the reference implementation with the following alterations. Instead of the BIDAF model, we use stronger BERT-BASE as the trained debiased model. Instead of using a TF-IDF-based bias model custom-tailored for a single bias type, we opt for a universal approach for obtaining biased models (Appendix B.2.1). We rerun the parameter search and use a different entropy penalty (H $= 0.4$) throughout all experiments.

Confidence Regularization (CREG) aims to reduce the model's confidence, i.e. the predicted score over samples marked as biased. Utama et al. (2020a) propose to reduce the confidence of the biased samples using a distillation from the conventional QA teacher model, scaled down by the relative scores of a biased predictor. In our experiments, we consistently use BERT-BASE for both the teacher and bias model. To enable comparability with LMIX, we use identical bias models for both methods (Described in Appendix B.2.1).

5 Results

Following the methodology introduced in §4, we assess the impact of selected training alterations of LLMs on Prediction bias and OOD performance.

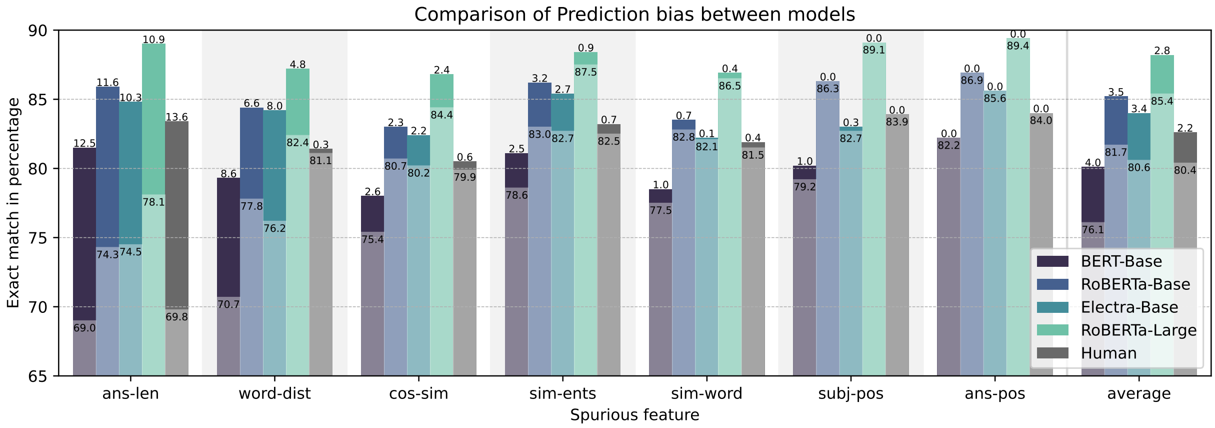

Figure 2: **Prediction bias per pre-trained model.** The worse-performing split performance (lower bars) and Prediction bias (upper bars, sorted by group average) of QA models trained from different pre-trained LLMs, trained and evaluated on SQuAD for Exact match. Per-group bootstrapping of 100 repeats with 800 samples.

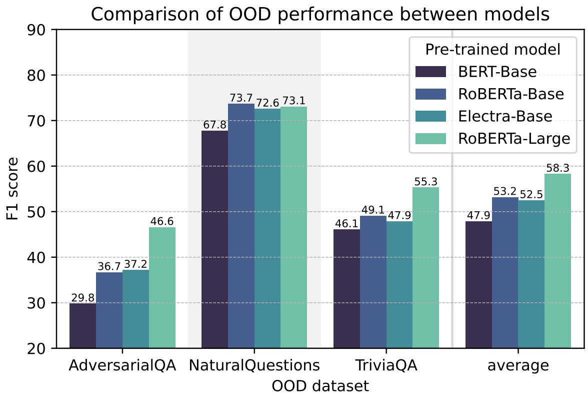

Figure 3: **OOD performance per pre-trained model.** Comparison of F1-score of different models fine-tuned on SQuAD and evaluated on listed OOD datasets.

5.1 Impact of Pre-training

Figure 2 compares the Prediction bias of the models using diverse pre-training data volumes and objectives. We observe that the selection of a base model results in differences in the scale of the fine-tuned model's Prediction bias.

The results suggest that increased amounts of pre-training data of the base models (cf. BERT-BASE and others) might mitigate the models' reliance on the bias. The results are less consistent in a comparison of different pre-training objectives (cf. ROBERTA-BASE and ELECTRA-BASE); While ELECTRA is less polarised in 4 out of 7 cases, the differences are minimal. The most significant gain presents an increase of the model size of ROBERTA-LARGE, reducing average Prediction bias by 1.2 points.

Analogically, Figure 3 compares OOD performance on selected QA datasets: AdversarialQA (Jia and Liang, 2017), NaturalQuestions (Kwiatkowski et al., 2019) and TriviaQA (Joshi et al., 2017). The average ranking is consistent with the conclusions of Prediction bias; increased pre-training data size improves the OOD performance, as well as the increase of the model size.

5.2 Prediction bias of OOD models

Figure 4 compares Prediction bias over the least-biased ROBERTA-LARGE models trained on different datasets. All evaluations are split on heuristics' thresholds T_h optimal for SQuAD model, which allows comparability to the shared human reference but implies that larger Prediction bias for OOD models might exist. We see that all Prediction biases learnt on SQuAD are also learnt from at least one OOD dataset. For Trivia model, all types of biases identified in SQuAD are magnified.

We specifically note the comparison of Prediction bias of the SQuAD model to the model trained on AdversarialQA, collected adversarially to a SQuAD model; We find that AdversarialQA model is the only OOD model lowering reliance on all biased features that are over the level of natural bias, supporting the argued efficiency of adversarial data collection in addressing original dataset biases.

5.3 Impact of Debiasing Methods

Figure 5 compares the biases of Question Answering models obtained using three debiasing methods (§4.3 – §4.4), applied to the most-biased BERT-BASE model. We observe that the methods are not consistent in the efficiency of mitigating the addressed bias feature. In fact, only RESAM baseline lowers the bias of the original model consistently. We attribute the inconsistency of debiasing meth-

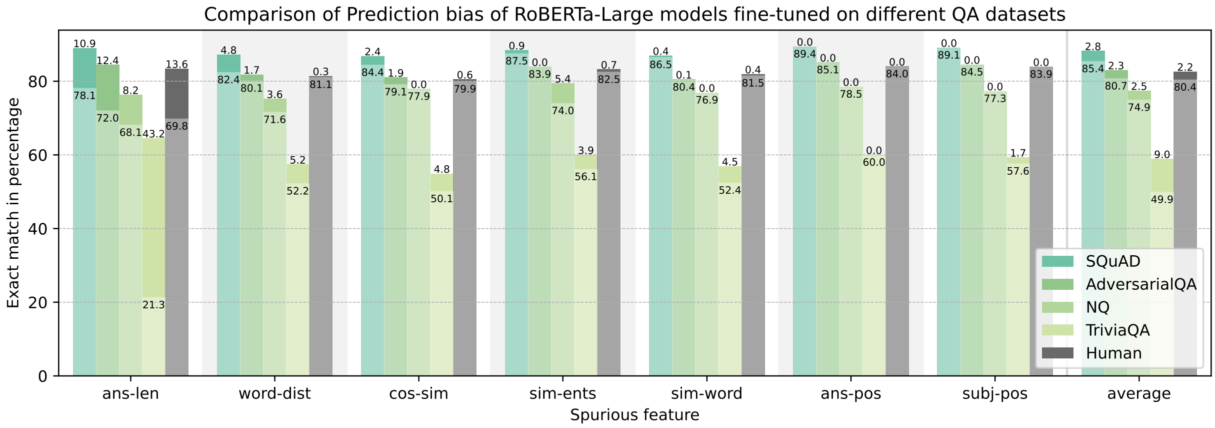

Figure 4: **Prediction bias per dataset.** The worse-performing split performance (lower bars) and Prediction bias (upper bars) of ROBERTA-LARGE trained on different QA datasets, evaluated on a validation split of SQuAD for Exact match. All evaluation splits are identical, identified as maximal for the SQuAD-trained model (Appx. C).

Table 1: **OOD performance of debiasing methods.** Differences of F1-scores of QA models trained on SQuAD using specified debiasing methods (§4.4) to address selected bias features (§4.1) evaluated on three OOD datasets; *AdversarialQA* / *NaturalQuestions* / *TriviaQA*, respectively. Top gains per dataset are in **bold**.

Original model:	29.8 / 67.8 / 46.1		
	ReSam	LMix	CReg
ans-len	−0.8 / −5.6 / −1.7	−0.9 / −19.7 / −3.3	−0.4 / +5.5 / +**2.1**
word-dist	+0.5 / +1.3 / +0.0	+0.9 / − 6.4 / +1.5	+**1.4** / +**7.5** / −0.5
cos-sim	−0.1 / +0.3 / −1.3	+0.4 / −11.3 / −4.1	−0.3 / +7.4 / +1.1
sim-ents	+1.1 / +1.5 / +0.3	−0.1 / −9.5 / −1.2	−1.0 / +5.9 / +2.0
sim-word	+0.3 / +0.1 / +0.4	−0.3 / −21.4 / −2.9	−0.7 / +3.9 / +1.4
subj-pos	−1.6 / −0.7 / −2.2	−1.3 / −14.8 / −1.3	+0.0 / +5.1 / +1.6
Average	−0.45	−5.31	+2.33

ods to their sensitivity to *bias model*, discussed in §6. While LMIX is the most efficient in addressing Prediction bias in standalone, consistently to Clark et al. (2019a), we see that often this feature comes for a price of the model's ID performance.

Table 1 enumerates the OOD performance of debiased models over three diverse QA datasets. By comparing the results to Figure 5, we see many cases, where improvements of OOD performance do not correspond to decays of Prediction bias; For instance, addressing *word-dist* bias using CREG improves OOD performance by 2.8% of an exact match on average and by 7.5 on *NaturalQuestions*, but the Prediction bias of such model increases by 1.1 points. A similar situation holds for CREG and *sim-word* bias, delivering 1.5-point average gain on OOD, but raising Prediction bias by 0.9 points.

Figure 6 additionally evaluates the impact of addressing one bias to other known biases in cases where each method delivers the largest Prediction bias reduction. We see that addressing a specific correlation also affects the scope of the model's reliance on other covariates. Results suggest that CREG might be more robust to a magnification of other biases, enlargening other Prediction biases by 0.31 on average, as compared to CREG (0.6) and RESAM (0.38).

6 Discussion

Impact of pre-training to models' robustness The bias-level analyses of diverse pre-trained models (Fig. 2) suggest that the mere increase of pre-training data and model parameters guide the fine-tuned models to lower reliance on biased features. However, we can find exceptions, such as in the case of ROBERTA-LARGE and ELECTRA-BASE on *ans-len*. We speculate that even larger volumes of data might make the model more attracted to taking a shortcut through easier problem formulations, such as through Named entity recognition (cf. BERT-BASE and ROBERTA-BASE on *sim-ents* bias). Out-of-distribution results (Fig. 3) aggregate the per-bias results, following the suggestive *bigger-data* and *bigger-model* rules. The average differences are comparable to debiasing techniques.

OOD performance and Prediction bias relation Our results conclude that the improvements of OOD performance attributed to the debiasing, reported by the previous work and in our experiments, might not be attributed to the mitigated reliance on a spurious correlation; (i) We measure that Prediction bias of the models trained directly on OOD datasets is still present over the level of human Prediction bias (§5.2). Therefore, it is possible to maintain OOD gains by learning to rely on bias fea-

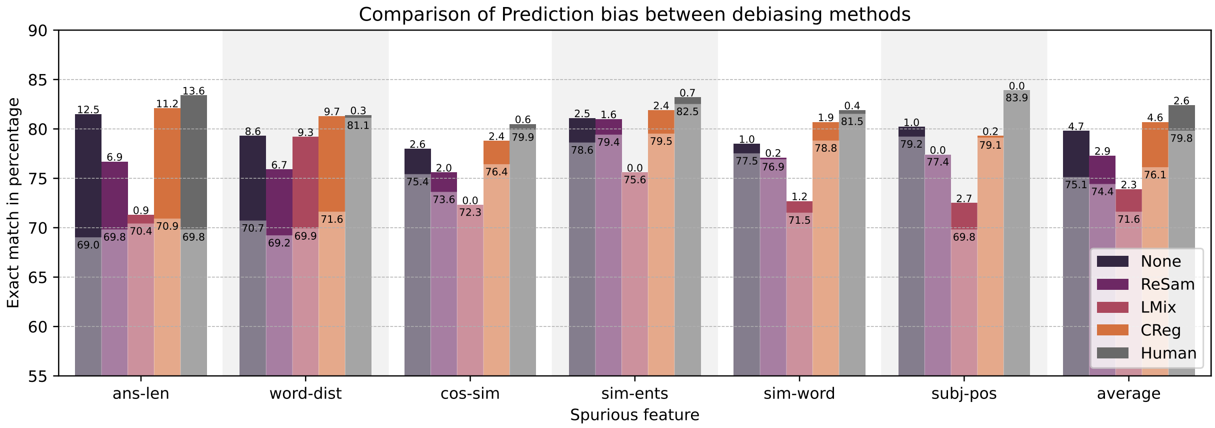

Figure 5: **Prediction bias per debiasing methods.** The worse-performing split performance (lower bars) and Prediction bias (upper bars) of BERT-BASE trained using selected debiasing methods, evaluated for Exact match on validation SQuAD. Per-group evaluations were measured using bootstrapping of 100 repeats with 800 samples.

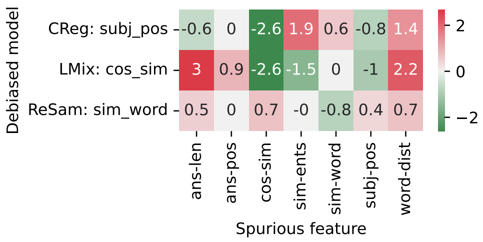

Figure 6: **Cross-bias evaluation of debiased models.** A relative change of Prediction bias by all spurious correlations, caused by applying inspected debiasing methods on BERT-BASE QA model, in addressing specified spurious correlation. A full matrix is in Appx. A, Fig. 7.

tures. (ii) In practice, we find cases where applying a debiasing method magnifies Prediction bias, but the resulting model still performs better on OOD, both on average and on specific datasets (§5.3).

Practical aspects of applying debiasing methods While we validate that debiasing methods enable improvements in the OOD, we find that the significance of such improvements largely varies between the addressed biases and the suitable configuration for one bias and dataset pair is often suboptimal for others. The scope of this variance can be seen in Table 1 from the comparison of average OOD performance of LMIX and CREG on *word-dist*, used to pick methods' hyperparameters and bias models (Appendix B.2), and other biases; Both of the methods perform best on the bias used in parameter tuning, and the differences are often large. Bias-specific parameter tuning is further convoluted by the speed of the convergence of debiasing methods, which we measure as approximately 4 times slower for CREG and 3.5 times slower for LMIX, compared to the standard fine-tuning of QA models.

The bias model is an important parameter of both assessed debiasing methods. We find that the scores have to be rescaled for trained bias models to avoid perplexing the trained model on biased samples and that the optimal scaling parameter is also bias-specific. The selection of the bias model also affects the optimal Entropy scaling H of LMIX; we find that the reported optimal value for AdversarialQA ($H = 2.0$) is also not close to optimal ($H = 0.4$) with our bias model.

7 Conclusion

This paper analyses the relationship between the model's learnt spurious correlations and out-of-distribution (OOD) performance, commonly used for the assessment of the robustness of LLMs. We build a simple framework to quantify models' prediction bias and analyze the impact of different pre-training and denoising strategies in addressing a number of known and novel biases of QA models.

We find many cases where state-of-the-art debiasing methods do not mitigate the model's reliance on a spurious correlation but still improve the OOD performance, suggesting that the inspected spurious features can be shared between ID and OOD datasets. We confirm this hypothesis by comparing the prediction bias of models trained on different datasets, showing that OOD models often rely on identical, spurious covariates as the ID model.

Our results motivate future work to more detailed assessments of reliance on specific, known biased features. Such assessments can allow future work to evade false conclusions on the covariates of models' robustness and foster progress toward reliable and socially unbiased language models.

References

Dimion Asael, Zachary Ziegler, and Yonatan Belinkov. 2021. A generative approach for mitigating structural biases in natural language inference. *arXiv preprint arXiv:2108.14006*.

Max Bartolo, A Roberts, Johannes Welbl, Sebastian Riedel, and Pontus Stenetorp. 2020. Beat the ai: Investigating adversarial human annotation for reading comprehension. *Transactions of the Association for Computational Linguistics*, 8:662–678.

Shikha Bordia and Samuel R. Bowman. 2019. Identifying and reducing gender bias in word-level language models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 7–15, Minneapolis, Minnesota. Association for Computational Linguistics.

Rui P. Chaves and Stephanie N. Richter. 2021. Look at that! BERT can be easily distracted from paying attention to morphosyntax. In *Proceedings of the Society for Computation in Linguistics 2021*, pages 28–38, Online. Association for Computational Linguistics.

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 2019a. 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.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019b. What does BERT look at? an analysis of BERT's attention. In *Proc. of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. ACL.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. *CoRR*, abs/2003.10555v1.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805v2.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proc. of the 2019 Conference of the NAACL: Human Language Technologies*, pages 4171–4186, Minneapolis, USA. ACL.

Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. 2018. Pathologies of neural models make interpretations difficult. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3719–3728, Brussels, Belgium. 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 Jeff Dean. 2015. Distilling the knowledge in a neural network. Cite arxiv:1503.02531Comment: NIPS 2014 Deep Learning Workshop.

Geoffrey E. Hinton. 2002. Training Products of Experts by Minimizing Contrastive Divergence. *Neural Computation*, 14(8):1771–1800.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing.

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.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

Rabeeh Karimi 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.

Divyansh Kaushik, Douwe Kiela, Zachary C. Lipton, and Wen-tau Yih. 2021. On the efficacy of adversarial data collection for question answering: Results from a large-scale randomized study. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6618–6633, Online. Association for Computational Linguistics.

Miyoung Ko, Jinhyuk Lee, Hyunjae Kim, Gangwoo Kim, and Jaewoo Kang. 2020. Look at the first sentence: Position bias in question answering. *arXiv preprint arXiv:2004.14602*.

Venelin Kovatchev, Trina Chatterjee, Venkata S Govindarajan, Jifan Chen, Eunsol Choi, Gabriella Chronis, Anubrata Das, Katrin Erk, Matthew Lease, Junyi Jessy Li, Yating Wu, and Kyle Mahowald. 2022. longhorns at DADC 2022: How many linguists does it take to fool a Question Answering model? A systematic approach to adversarial attacks. In *Proceedings of the First Workshop on Dynamic Adversarial Data Collection*, pages 41–52, Seattle, WA. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, M. Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR*.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference. In *Proc. of the 57th Annual Meeting of the ACL*, pages 3428–3448, Florence, Italy. ACL.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2021. A Survey on Bias and Fairness in Machine Learning. *ACM Comput. Surv.*, 54(6).

Yulei Niu, Kaihua Tang, Hanwang Zhang, Zhiwu Lu, Xian-Sheng Hua, and Ji-Rong Wen. 2021. Counterfactual VQA: A Cause-Effect Look at Language Bias. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 12700–12710. Computer Vision Foundation / IEEE.

Yulei Niu and Hanwang Zhang. 2021. Introspective distillation for robust question answering. In *Advances in Neural Information Processing Systems*, volume 34, pages 16292–16304. Curran Associates, Inc.

Alicia Parrish, Angelica Chen, Nikita Nangia, Vishakh Padmakumar, Jason Phang, Jana Thompson, Phu Mon Htut, and Samuel Bowman. 2022. BBQ: A hand-built bias benchmark for question answering. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2086–2105, Dublin, Ireland. Association for Computational Linguistics.

Alec Radford and Karthik Narasimhan. 2018. Improving Language Understanding by Generative Pre-Training.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proc. of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, USA. ACL.

Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. 2020. Distributionally robust neural networks. In *International Conference on Learning Representations*.

Kazutoshi Shinoda, Saku Sugawara, and Akiko Aizawa. 2021. Can question generation debias question answering models? a case study on question-context lexical overlap. *arXiv preprint arXiv:2109.11256*.

Michal Štefánik, Vít Novotný, Nikola Groverová, and Petr Sojka. 2022. Adaptor: Objective-Centric Adaptation Framework for Language Models. In *Proceedings of the 60th Annual Meeting of the ACL: System Demonstrations*, pages 261–269, Dublin, Ireland. ACL.

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 ACL*, 8:621–633.

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 *Proc. of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7597–7610. ACL.

Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, Morgan Funtowicz, Joe Davison, Sam Shleifer, et al. 2020a. 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.

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. 2020b. Transformers: State-of-the-Art Natural Language Processing. In *Proc. of the 2020 Conf. EMNLP: System Demonstrations*, pages 38–45. ACL.

Mingzhu Wu, Nafise Sadat Moosavi, Andreas Rücklé, and Iryna Gurevych. 2020. Improving QA generalization by concurrent modeling of multiple biases. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 839–853. Association for Computational Linguistics.

Ruibin Xiong, Yimeng Chen, Liang Pang, Xueqi Cheng, Zhi-Ming Ma, and Yanyan Lan. 2021. Uncertainty calibration for ensemble-based debiasing methods. In *Advances in Neural Information Processing Systems*.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase Adversaries from Word Scrambling. In *Proc. of the 2019 Conf. NAACL-HLT*, pages 1298–1308, Minneapolis, USA. ACL.

Chunting Zhou, Xuezhe Ma, Paul Michel, and Graham Neubig. 2021. Examining and combating spurious features under distribution shift. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12857–12867. PMLR.

A Cross-Bias Matrix of All Debiased Models

Figure 7 shows the change of Prediction bias by applying the listed debiasing methods to eliminate the associated bias feature. We see that some biases are more difficult to address, while other ones can be transitively addressed through others.

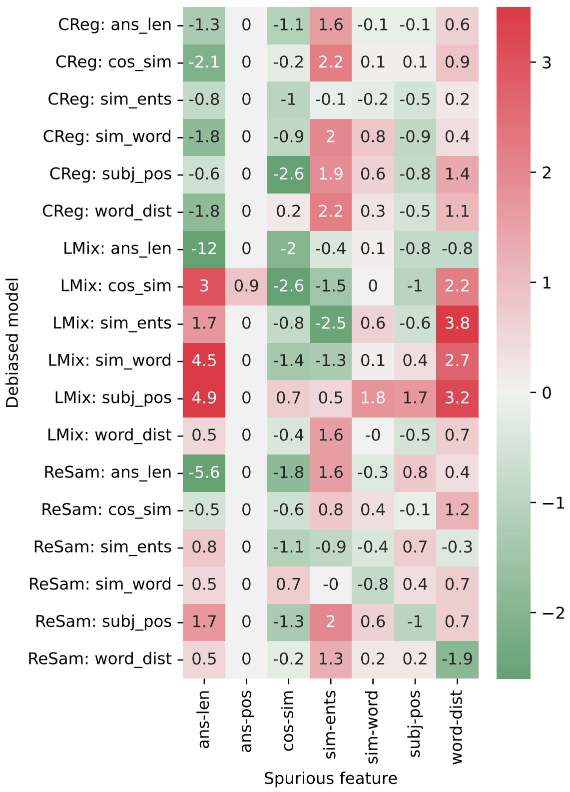

Figure 7: **Full cross-bias evaluation of debiased models.** A relative change of Prediction bias by all spurious correlations, caused by applying inspected debiasing methods on BERT-BASE QA model, in addressing specified spurious correlation.

B Details of Training Configurations

This section overviews all configurations that we have set in training the debiased models (§4.3 – 4.4) as well as the conventional QA fine-tuning comparing the impact of pre-training on QA models' robustness (§4.2).

B.1 Standard Fine-tuning

For model fine-tuning, we use following hyperparameters: **learning rate:** $2e^{-5}$, **batch size:** 16, **evaluation:** each 200 steps and **train epochs:** 3. We also set the **early stopping patience** to 10 evaluation steps, based on a validation loss of the training dataset (SQuAD) also used for selecting the evaluated model. The **validation loss** of the evaluated model is 1.02. All other parameters can be retrieved from the defaults of TrainingArguments of HuggingFace (Wolf et al., 2020b) in version 4.19.1.

B.2 Debiasing Training Experiments

B.2.1 Bias models

The canonical debiasing implementations utilize bias-specific models for identifying bias; Clark et al. (2019b) use the TF-IDF model as a scalar of possible bias for each QA sample, while Utama et al. (2020a) experiment with a percentage of the shared words and cosine embeddings between word distances, in NLI context.

As we scale our experiments to six different biases, we opt for a universal approach for obtaining bias models for both LMIX and CREG and train each bias' model on a better-performing segment of the dataset identified using the approach described in Section 3. For all our biased models, we train BERT-BASE architecture from scratch and pick the checkpoint with a maximal difference of the F1-score between the two segments from the validation split of SQuAD.

While our approach scales well over many biases, a significant difference between the learned bias models original ones, such as TF-IDF, is the *scale* of prediction probabilities; As the trained bias models become very confident on a biased subset, often reaching probabilities close to 1 for the biased samples. A "perfect" bias model causes problems for both LMIX and CREG as such model forces the trained model to avoid correct predictions on the biased samples completely. We learn to address this problem by rescaling bias predictions and tuning the scaling interval based on a validation performance of the debiased model. Consequently, we scale the bias probabilities to $\langle 0; 0.2\rangle$ for LMIX and $\langle 0; 0.1\rangle$ for CREG. Further details on bias models can be found in Appendix B.2.

In the initial phase, we experiment with diverse configurations and sizes of bias models, intending to maximize the polarization of performance on the biased and non-biased subsets. Among different configurations of model sizes and configurations, we find that the highest polarisation can be reached using BERT-BASE architecture trained from scratch. We fix this decision and the parameters (learning rate 4e^{-5}, a number of training steps 88,000) with respect to the maximum OOD (AdversarialQA) F-score of this model of LMIX model addressing *word-dist* bias. Our bias models reach between 18% and 59% of accuracy on easier, i.e., biased data split while between 4% and 19% on the non-biased one.

B.2.2 Baseline debiasing: Resampling

We train the RESAM analogically to Baseline Fine-tuning experiments (§B.1). Compared to other debiasing methods, RESAM baseline is non-parametric, including no dependence on the bias model.

Even though we find RESAM to be the only method mitigating Prediction bias in all the cases, our further analyses show that its enhancements on OOD datasets vary among biases. Figure 8 shows validation losses from the training on SQuAD resampled using RESAM by *word-dist*, while analogically, Figure 9 shows the losses for *sim-ents* bias. While in the former case, RESAM does not stably reach lower loss on OOD datasets, in the latter case, validation losses are consistently lower between steps 7,000 and 8,000, where the SQuAD validation loss used to pick the best-performing model plateaus.

B.2.3 Learned Mixin

In addition to the implementation and default parameters of Clark et al. (2019a), we find that the additional entropy regularization component H makes a significant difference in the resulting model evaluation. Therefore we perform a hyperparameter search over the values of H used for QA by Clark et al. (2019a) on *word-dist* bias, optimizing the OOD performance on AdversarialQA (Bartolo et al., 2020) and eventually fix $H = 0.4$ over all our experiments.

Following the low initial OOD performance of LMIX as compared to the results of Clark et al. (2019a), we further investigate covariates of this result and identify LMIX's high sensitivity to bias model; while in the original implementation, TF-IDF similarities of question and answer segment likely never reach 1.0, our generic bias models reaches 1.0 probability for most of the samples marked as biased. Hence, we introduce a parameter of scaling interval $\langle 0; x\rangle$ of bias model's scores, where we optimize $x \in \langle 0.2; 0.4; 0.5; 0.6; 0.7; 0.8; 0.9; 0.95\rangle$ according to the maximum ID F-score of the debiased model addressing *word-dist* bias, fixing optimal $x = 0.8$ throughout all other experiments. All other parameters remain the identical to the standard fine-tuning (§B.1).

We implement LMIX using Adaptor library (Štefánik et al., 2022) in version 0.1.6.

B.2.4 Confidence Regularization

While the authors of CREG (Utama et al., 2020a) find benefits in its non-parametricity, we find that CREG also shows high sensitivity to a selection of bias model, guiding us to also rescale the prediction of the bias model in the training distillation process. We use the same methodology to pick the scaling interval $\langle 0; x\rangle$ for CREG as for LMIX and fix $x = 0.9$ as the optimal one. All other parameters remain the identical to the standard fine-tuning (§B.1).

We implement CREG using Transformers library (Wolf et al., 2020a) in version 4.19.1.

C Exploiting Heuristics Configuration

Here we enumerate the optimal thresholds over all pairs of the implemented heuristics, as picked according to BERT-BASE-CASED model.

We assess the candidate thresholds among all possible values within the range of the computed values A_h computed over X = SQuAD$_{\text{valid}}$ (see Algorithm 1), with steps of 1 for possible values higher than 1 and 0.1 for values between 0 and 1, within the valid interval; We set the validity interval such that the resulting splits of the dataset must each have a size of at least two times of the sample size parameter, except where there is only one significant threshold, and its size is larger than the sample size. The optimal threshold value is then the one that delivers the highest Prediction bias value. We find and use the following optimal thresholds of BERT-BASE-CASED evaluated on

X = $\text{SQuAD}_{\text{valid}}$ for specific biases: 7 for *word-dist*, 3 for *sim-word*, 4 for *ans-len*, 0.1 for *cos-sim*, 0 for *sim-ents* and 1 for *subj-pos*. A corresponding number of samples in the underperforming groups of $\text{SQuAD}_{\text{valid}}$ (n=10,570) are following: 1,651 for *word-dist*, 3,281 for *sim-word*, 3,124 for *ans-len*, 954 for *cos-sim*, 5,006 for *sim-ents* and 1,672 for *subj-pos*.

The implementations of some biases' heuristics utilize external libraries for entity recognition or TF-IDF vectorization. For these, we used SpaCy in version 3.4.1 and NLTK in version 3.4.1.

D Experimental Environment

Our experiments utilized a single NVidia A100 GPU with 80 GB of VRAM, a single CPU core, and less than 32 GB of RAM. However, all our experiments can be run using a lower compute configuration, given a longer compute time; The inference of a single-sample prediction batch of RoBERTa-Large as our largest model requires only 13 GB of VRAM. The debiasing training runs take longer to converge, as compared to standard fine-tuning; While the conventional training and ReSam converges within 10,000 steps (Figures 8 and 9) we find that LMix requires between 60,000 and 100,000 steps, and CReg needs between 20,000 and 30,000 steps to converge, making the debiasing training 4–8 times slower in average. In our training configuration, each of the reported training runs takes between 50 minutes and 1 hour per 10,000 updates. Given that our evaluation already aggregates the bootstrapped results, we perform a single run for each experiment, which might result in a wider confidence interval and consistently smaller measured volumes of Prediction bias.

Figure 8: Development of validation loss of **ReSAM** addressing ***word-dist*** bias (darker plots) and standard fine-tuning (lighter plots) for Question Answering on SQuAD, also evaluated on other (OOD) datasets, for the first 10,000 steps.

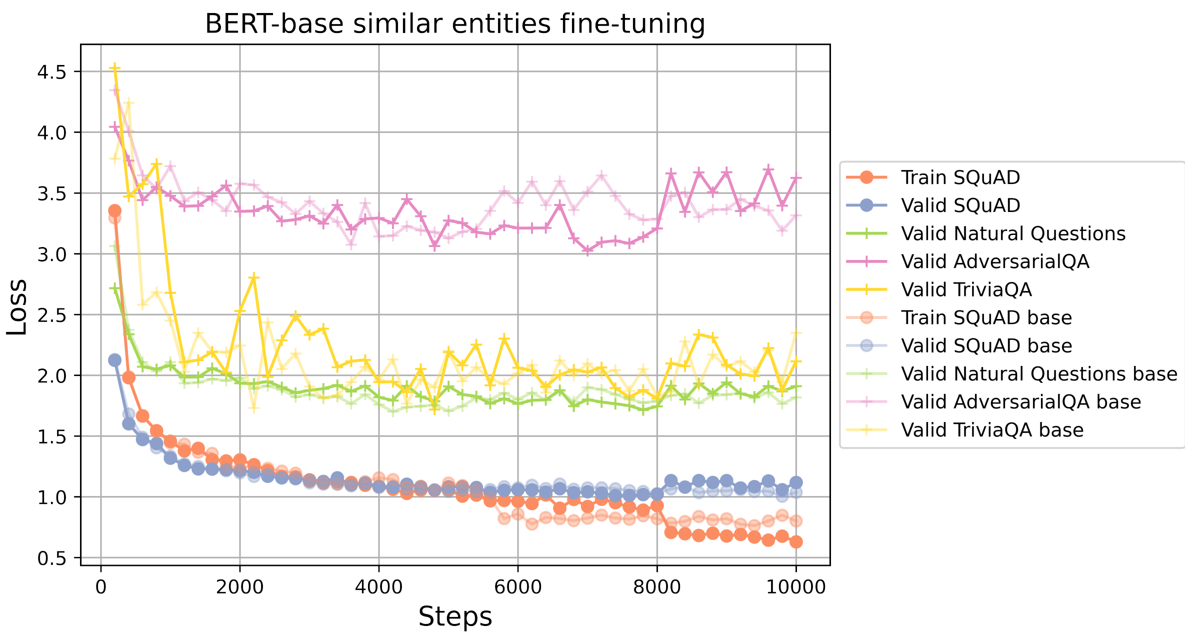

Figure 9: Development of validation loss of **ReSAM** addressing ***sim-ents*** bias (darker plots) and standard fine-tuning (lighter plots) for Question Answering on SQuAD, also evaluated on other (OOD) datasets, for the first 10,000 steps.