Think Twice: Measuring the Efficiency of Eliminating Model Biases in Question Answering

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 biases 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 introduce a framework for finer-grained analysis of discovered model biases and measure the significance of some previously-reported biases while uncovering several new ones. The bias-level metric allows us to assess how well different pre-trained models and state-of-the-art debiasing methods mitigate the identified biases in Question Answering (QA) and compare their results to a resampling baseline. We find cases where bias mitigation hurts OOD performance and, on the contrary, when bias enlargement corresponds to improvements in OOD, suggesting that some biases are shared among QA datasets and motivating future work to refine the analyses of LLMs' robustness.

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).

We jointly refer to these phenomena with a term of *bias*; *dataset biases*, i.e. largely-valid, yet non-representative predictive patterns present in the training set are commonly fingerprinted in the model during the training, creating a *model bias*.

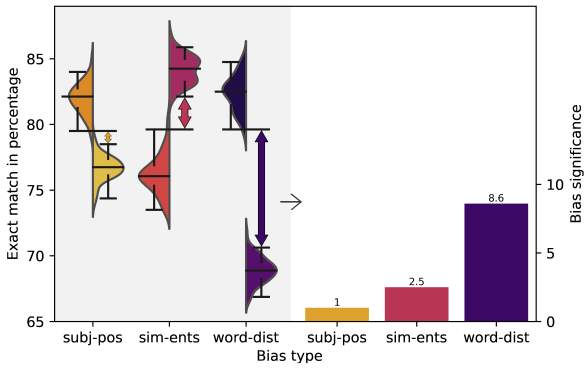

Figure 1: We quantify model bias using bootstrapped evaluation on segments of data separated by exploiting chosen bias (left) and subsequently, by measuring the difference in model's performance over these two groups (right), that we refer to as *Bias significance* (§3).

Arguably, a major motivation for eliminating models' biases is to enhance their robustness in practical deployments, avoiding a decrease in quality 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). While such estimates provide information on model robustness under a specific domain shift, the OOD datasets might share some of the training, in-distribution (ID) biases, conditioned, for instance, by data collection methodology or human annotators' background (Mehrabi et al., 2021), conversely allowing to reach higher OOD performance by modeling shared bias. Additionally, using only the OOD evaluation, the robustness of the model to the adversarial samples exploiting specific biases remains hidden, leaving their practical deployments vulnerable to systematic errors.

Our work addresses this gap, presenting a framework to quantify model bias by comparing models' performance on data segments split on an attribute exploiting the bias. We apply this methodology to measure the biases of selected commonly-used LLMs for extractive QA. We report on the significance of several previously identified biases and some new ones that we identify as significant. Finally, we assess the efficiency of the state-of-the-art debiasing methods and a resampling baseline in eliminating these biases and evaluate the significance of model bias and its OOD performance.

We find that the relation between model bias and OOD performance is not straightforward; While the debiasing methods can largely mitigate the addressed model bias, often such-debiased models do not reach gains in the model's OOD performance. Conversely, a magnification of bias can correspond to large OOD gains. Our findings suggest that many biases might be shared among the datasets and motivate the presented practice of assessing robustness not only by OOD performance but also from the perspective of specific, known biases.

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 bias significance. 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 biases, and the design of specific heuristics that 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.

2 Background

Biases of NLP datasets Previous work analyzed erroneous subsets of LLMs' test sets and identified numerous false presumptions 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. Poliak et al. (2018) show that the NLI model might even learn to ignore the text of the premise and draw conclusions solely on the hypothesis. Asael et al. (2021) identify model's sensitivity to meaning-invariant structure permutations. Similarly, Chaves and Richter (2021) identify BERT's reliance on invariant morpho-syntactic composition of the input.

In the extractive Question Answering task, identifying an answer to a given question in the given context, LLMs often rely on mutual positional information of the question and possible answer words. Jia and Liang (2017a) show that QA models learn to rely on the proximity of the answer to the words shared between the question and context. Bartolo et al. (2020) find that instead of learning to reason the answer, models tend to identify the question keywords and look for the passages containing similar keywords, remaining vulnerable to samples with none or multiple occurrences of the keywords in the context. Ko et al. (2020) demonstrate models' inclination to give answers using only the first two sentences of the context, being statistically most likely to answer human-created questions.

A perspective direction circumventing the biases conditioned by data collection is presented in adversarial data collection (Jia and Liang, 2017a; 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. Such behavior suggests that adversarial collection might bring its own set of biases, of which some were already identified in the QA domain (Kovatchev et al., 2022).

Debiasing methods Other works address the documented biases of the NLP model through debiasing methods, mitigating one or more biases of LLMs. 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, followed by (ii) training the debiased model as a complement to the biased one, as an application of the model-agnostic 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. Moving away from addressing a single bias, Niu and Zhang (2021) simulates the model for non-biased, out-of-distribution dataset through coun-

terfactual reasoning (Niu et al., 2021) and uses the resulting distribution in a weighted distillation (Hinton et al., 2015), similarly to the LearnedMixin. Utama et al. (2020b) and Wu et al. (2020) also omit the assumption of the bias' knowledge and presume a representation of bias in the overconfidence of the general model.

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 biased model. Consistently to our experience, Xiong et al. (2021) argue that a more precise calibration of the biased model might bring further benefits to this framework.

Measuring model bias Most of the referenced work evaluates the acquired robustness on OOD datasets, i.e. samples of the same task but collected independently. In some cases, the evaluation utilizes OOD datasets specifically 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 the *biased* and *non-biased* subsets and compared model behavior between the two groups. McCoy et al. (2019) perform such evaluation over the biases of MNLI, demonstrating large margins in accuracy over the two groups and superior robustness of BERT over previous models. Utama et al. (2020b) compares model confidence between the two groups. Our bias measure elaborates further in this direction, providing a statistical assessment of the model performance polarisation over such groups.

3 Measuring Bias Significance

To provide a fair and reliable comparison of different models for their bias, we formalize a technique based on a comparison of two segments of data divided by a heuristic exploiting a chosen bias.[1] The following steps of the proposed bias assessment are also described in Algorithm 1 and visualized in Figure 1.

We begin 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 that we suspise as non-representative, yet predictive for our end task and hence, possibly relied upon 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 Θ on both of these segments, and (vi) measure the *Bias significance* as the difference in performance between the 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.

Our approach presumes that the performance of the biased model can be significantly polarised by picking samples where a simple heuristically-exploitable attribute does or does not hold. As such, one should note that such measure should not be used in a standalone but rather as a complement to OOD evaluations since the sole bias significance is reduceable merely by lowering the performance on the biased, better-performing subset. Additionally, even though we perform a hyperparameter search for T_h feasible for a given size of dataset splits, no guarantees on the maximality of the correspond-

func *measure_bias*(Θ, 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*$(\Theta\,(X_1'))$
  foreach $X_2' \in$ *repeat*(*sample*(X_1)) **do**
    $E_2 \leftarrow E_2 +$ *evaluate*$(\Theta\,(X_2'))$
  $dist \leftarrow \max(0;\, E_1^{\downarrow} - E_2^{\uparrow};\, E_2^{\downarrow} - E_1^{\uparrow})$
  return $dist$

Algorithm 1: We measure *Bias significance* of the model Θ exploited by the *heuristic* h on dataset X, as a *difference* of Θ'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 such distance a *significance* of the exploited bias.

[1] Implementation of our Bias significance measure will be available on a GitHub link here.

ing polarisation can be obtained. Hence our Bias measurement technique only provides the *lower bounds* of the model's worst-case polarisation.

4 Experiments

Our experiments assess the significance of known biases of LLMs and the impact of selected alterations in the training configuration on the scales of these biases in the resulting models. Given a large body of previous work documenting biases of Question Answering models, we specifically focus on QA. For all the documented and newly-identified biases of QA models, we first describe and implement the exploiting heuristics we use to measure the Bias significance (§4.1). Subsequently, we observe the impact of the selected pre-training strategies (§4.2) and of selected debiasing methods (§4.3 – §4.4) on the bias significance and OOD performance of QA models.

4.1 Biases and Exploiting Heuristics

Following is a list of biases of QA models that we evaluate in our experiments. While a majority of these biases are either introduced or mentioned in the previous work, we further extend this list using our empirical experience with two novel biases that we later assess as significant. The biases introduced in this work are preceded with **+**.

Together with each bias, we also briefly describe its exploiting heuristic computing the possibly-predictive bias attribute A_h (Algorithm 1).

Distance of Question words from Answer words (*word-dist*) Jia and Liang (2017a) propose that the models are prone to return answers close to the vocabulary of the question in context. The corresponding heuristic 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 the exploiting 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 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 the models trained on SQuAD (Rajpurkar et al., 2016) are biased by the relative position of the answer as a consequence of the over-representations of the samples with answers in the first sentence. Our exploiting heuristic segments the context into sentences first, identifies the sentence containing the first answer, and yields a scalar representing the ordering of the sentence within the context that contains the answer (A_h).

Cosine similarity of TF-IDF representations between Context and Question (*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 bias in the corresponding heuristic by fitting the TF-IDF model on all SQuAD contexts, used to infer the TF-IDF vectors of questions and their corresponding answers, 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 by presenting QA models trained on SQuAD with samples of significantly higher answer lengths, the QA models yield correct spans much less frequently. This observation implicitly identifies models' reliance on the irrelevant presumption that the answer must comprise at most a few words. We exploit this bias by simply computing A_h as the length of the answer. In the cases where multiple, diverse answers are considered valid, we average their lengths.

+Number of Answer's named entities in Context (*sim-ents*) Based on our observations, we suspect that the in-context presence of multiple named entities of the same type, such as multiple personal names or locations, might impact the QA model's prediction quality. This might suggest that models tend to reduce the QA task to a simpler yet not fully relevant problem of Named Entity Recognition. To exploit such bias, we utilize a pre-trained BERT model provided within SPACY library (Honnibal and Montani, 2017) to identify named entities in specific answers and contexts. If no entity resides within the question, we return $A_h = 0$; otherwise, we count A_h as the number of named entities of the same type as in the question in the context. If the question contains multiple entities of a different type, we return the maximum over all types.

+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 identify the subject from the question and its position in the context. Then we locate the answer in the context 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

With no alterations to the traditional approach for training LLMs for QA (Devlin et al., 2019), we fine-tune a set of diverse pre-trained LLMs, estimating the impact of the selection of the pre-trained model on the robustness of the final QA model given by its Bias significance and OOD performance.

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 super-sampling 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

To assess the efficiency of the current debiasing methods in mitigating the addressed biases, we implement and evaluate the representatives of two diverse debiasing methods and measure their impact on Bias significance and OOD performance, following the measurement methodology described in Section 3. We follow the reference implementations[2] and highlight the significant alterations in the following description. A complete description of our training settings can be found in Appendix A.2.

LearnedMixin (LMIX) method introduced by Clark et al. (2019b) is a popular adaptation of Product-of-Experts framework (Hinton, 2002), followed by diverse refinements (Section 2) and uses a biased model as a complement of the trained debiased model in a weighted composition. We use the reference implementation and its parameters with the following alterations. Instead of the BIDAF model, we use 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 (§4.4.1). We also rerun the parameter search and use a different entropy penalty $H = 0.4$ throughout the experiments.

Confidence Regularization (CREG) aims to reduce the model's confidence, i.e. the predicted score over samples presumed to be biased. One of the latest instances of this direction is a debiasing strategy of Utama et al. (2020a), that refines the reduction of the bias from the predicted scores using 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 (§4.4.1).

4.4.1 Bias models

The original 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 seven different biases, we opt for a universal approach for obtaining bias models for both LMIX and CREG and train each bias' model on a *biased* 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

[2]We will make all bias heuristics, biased and debiased models available for future experiments under a link here.

the biased and non-biased segment of the validation split of SQuAD.

While our approach scales well over many biases, a significant difference between the learned bias models and the unsupervised bias models, such as TF-IDF, is the scaling of prediction probabilities; As the trained bias models become gradually more confident on a biased subset, they often reach 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 training our bias models can be found in Appendix A.2.

5 Results

Following the methodology introduced in §4, we assess the impact of selected training alterations of LLMs on Bias significance and OOD performance of resulting models.

5.1 Impact of Pre-training

Figure 2 compares the Bias significance 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 significance of the fine-tuned model's 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 Bias significance by 1.2 points.

Analogically, Figure 3 compares OOD performance on selected QA datasets: AdversarialQA (Jia and Liang, 2017b), NaturalQuestions (Kwiatkowski et al., 2019) and TriviaQA (Joshi et al., 2017). While the relative ranking of the models varies between the datasets, the average ranking is consistent with the conclusions of Bias significance; increased pre-training data size might also help the OOD performance, as well as the increase of the model size.

5.2 Impact of Debiasing Methods

Figure 4 compares the biases of Question Answering models obtained using three debiasing methods (§4.3 – §4.4), applied to BERT-BASE-CASED.

We observe that the methods are not consistent in the efficiency of mitigating the addressed Bias significance, and there is not a single method that is more efficient than the others. While LMix is the only method mitigating *word-dist* and *cos-sim* biases below the significance level, it underperforms on *subj-pos* bias and might even enlarge the bias of the original model, as in the case of *sim-ents*. In fact, only RESAM baseline consistently lowers the bias of the original model. We attribute inconsistency in the efficiency of debiasing methods to their high sensitivity to properties of the *bias* models, which we discuss further, in Section 6.

Table 1 enumerates the OOD performance of debiased models over three diverse QA datasets. Similarly to Bias significance, there is no clear cut-off of the most efficient configuration for any dataset. Interestingly, it is difficult to observe correspondence in success in the bias mitigation (Figure 4) to the OOD results. While the most robust model by OOD performance is obtained by addressing *word-dist* bias using CREG, improving OOD performance by 2.8% on average and by 7.5 on *NaturalQuestions*, the Bias significance of such model counterintuitively increases by 1.1 points compared to the standard QA model. Similar situation holds for CREG and *sim-word* bias, delivering 1.5-point average gain on OOD, but raising bias significance by 0.9 points. However, the inverse scaling of bias significance and OOD performance does not hold for all biases; For instance, addressing *subj-pos* bias with CREG brings OOD improvement of 2.3% and a decay of Bias significance by 0.8.

Addressing some biases, on average, delivers higher OOD gains than others. The most efficient biases to address are *word-dist*, improving OOD in 6 cases, or *sim-ents* and *sim-word* in 5 cases, while as the least efficient ones seem *ans-len*, helping in 2 cases or *subj-pos* in 3 cases. While it is tempting to conclude which biases are shared between SQuAD and OOD datasets from this enumeration, we note that the results are conditioned by the relatively variable efficiency of the debiasing methods over different biases.

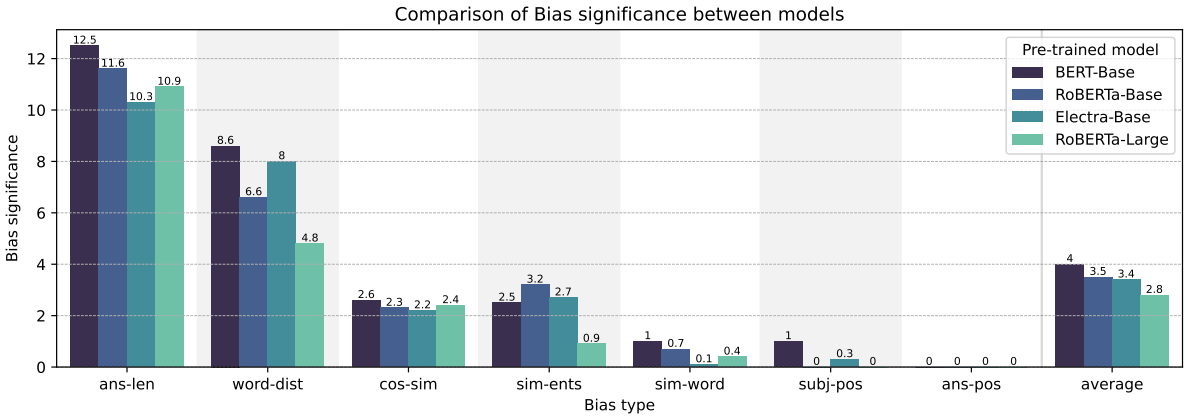

Figure 2: **Bias significance per pre-trained model.** Comparison of Bias significance of QA models trained from different pre-trained LLMs. Per-group results were measured using bootstrapping of 100 repeats with 800 samples.

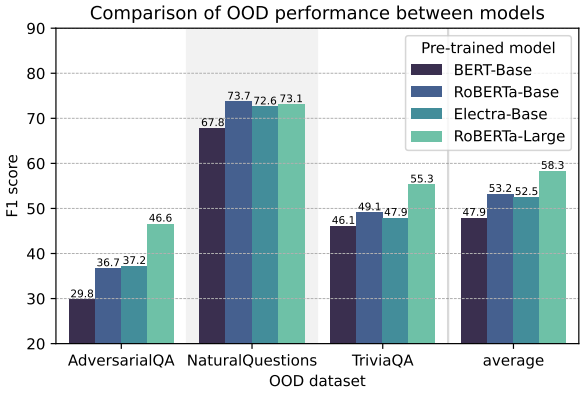

Figure 3: **OOD performance per pre-trained model.** Comparison of F1-score of models fine-tuned from selected LLMs, evaluated on listed OOD datasets.

6 Discussion

Impact of pre-training to models' robustness The bias-level analyses of the QA models fine-tuned from diverse pre-trained models (Figure 2) suggest that the mere increase of data and model parameters does not lead to a complete avoiding of non-representative heuristic shortcuts from the models' decision-making but still guides the fine-tuned model to lower reliance on biased features of the problem. While such ranking holds on average, even larger models might not necessarily be more resistant to specific biases, such as in the case of RoBERTa-Large and Electra-Base and the most-significant bias of Answer length. We speculate that even larger volumes of data might make the model more attracted to taking the 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 seem to aggregate this property more strictly. Figure 3 shows minimal differences in the ranking of the fine-tuned models over different OOD datasets. Subsequently, improvements in the average follow the suggestive *bigger-data* and *bigger-model* rules, with the average differences similar to debiasing techniques.

Relation of OOD performance and model's bias Most of the previous work on debiasing LLMs demonstrates the efficiency of debiasing by improvements on a chosen OOD dataset (§2). While such evaluations are valuable for possible applications within the evaluation domain, our results suggest that often such OOD improvements might not be attributed to the bias elimination; we find cases where a bias polarises the debiased model's performance even more than the original model, but the model improves OOD performance both in average and on specific datasets (§5.2).

We argue that increases in both Bias significance and OOD performance might be attributed either to the bias being shared between the training and evaluation dataset or to the existence of the *inverse* bias within the evaluated OOD dataset, where the dataset can be heuristically exploited by following the *inverse rule* to the rule valid in the training dataset. Such a situation appears viable in the context of the adversarial data collection of AdversarialQA, where the samples are collected with an explicit aim to mislead the SQuAD QA model.

Limited conclusions can be drawn in the cases where the OOD performance does not improve; Even in cases of reduced Bias significance, the model's performance can be corrupted as a side-product of the debiasing process. While we can not

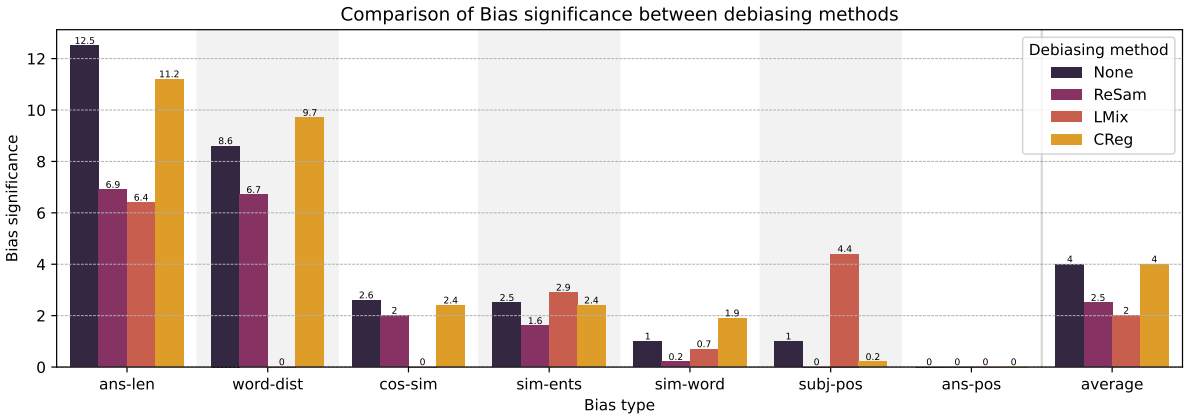

Figure 4: **Bias significance per debiasing methods.** Comparison of Bias significance between debiasing methods, applied in debiasing BERT-BASE model. Per-group evaluations were measured using bootstrapping of 100 repeats with 800 samples.

reject that the bias elimination caused the decay of the performance on a given OOD dataset, implying that the bias is shared, we instead propose to evaluate the biases of the model trained directly on such OOD dataset, if permitted by its size.

Practical aspects of applying debiasing methods Even though we confirm that debiasing methods enable improvements in the OOD performance of LLMs, we find that the significance of such improvements largely varies between the addressed biases and that 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 addressing *word-dist*, used to pick methods' hyperparameters and bias models (Appendix A.2), to 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 8 times slower for LMIX, compared to the standard fine-tuning of QA models.

The bias model is an important parameter of both assessed debiasing approaches. We find that the scores have to be rescaled for highly confident bias models to avoid perplexing the debiased models 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.

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 biases overviewed in §4.1 and evaluated on a selection of OOD datasets; *AdversarialQA* / *NaturalQuestions* / *TriviaQA*, respectively. The largest 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	−12.2 / −24.3 / −2.3	−0.4 / +5.5 / +2.1
word-dist	+0.5 / +1.3 / +0.0	− 5.0 / − 3.4 / **+7.3**	**+1.4** / **+7.5** / −0.5
cos-sim	−0.1 / +0.3 / −1.3	−16.0 / −28.9 / −4.2	−0.3 / +7.4 / +1.1
sim-ents	+1.1 / +1.5 / +0.3	− 8.7 / −19.5 / −0.2	−1.0 / +5.9 / +2.0
sim-word	+0.3 / +0.1 / +0.4	−15.0 / −34.6 / −8.1	−0.7 / +3.9 / +1.4
subj-pos	−1.6 / −0.7 / −2.2	−16.2 / −32.3 / −7.1	+0.0 / +5.1 / +1.6
Average	−0.45	−12.82	+2.33

7 Conclusion

This paper analyses the relationship between the model's prediction bias and out-of-distribution performance, commonly used to assess 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 diverse set of documented and newly-found biases of QA models.

We find empirical evidence for our initial hypothesis that bias mitigation does not always correspond to enhancements of the model's OOD performance, suggesting that many of the inspected biases are shared between ID and OOD datasets. Our results strive to motivate future research enhancing the robustness of LLMs to more detailed assessments of bias alterations that may allow future work to evade false conclusions on the covariates of models' robustness, fostering steady progress toward more reliable deployments of LLMs.

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.

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. *arXiv preprint arXiv:1909.03683*.

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. To appear.

Robin Jia and Percy Liang. 2017a. Adversarial examples for evaluating reading comprehension systems. *ArXiv*, abs/1707.07328.

Robin Jia and Percy Liang. 2017b. Adversarial examples for evaluating reading comprehension systems. *arXiv preprint arXiv:1707.07328*.

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.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis Only Baselines in Natural Language Inference. In *Proc. of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, USA. ACL.

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.

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.

A 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).

A.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.

A.2 Debiasing Training Experiments

A.2.1 Bias models

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 fix the learning rate (4e-5), and a number of steps (88 000) with respect to the maximum OOD (AdversarialQA) F-score of this model in LMIX addressing *word-dist* bias. Our bias models reach between 18%, and 59% of accuracy on bias data split while between 4% and 19% on the non-biased one.

A.2.2 Baseline debiasing: Resampling

We train the RESAM analogically to Baseline Fine-tuning experiments (§A.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 bias significance in all the cases, our further analyses show that its enhancements on OOD datasets vary among biases. Figure 5 shows validation losses from the training on SQuAD resampled using RESAM by *word-dist*, while analogically, Figure 6 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.

A.2.3 Learned Mixin

In addition to the implementation and default parameters of Clark et al. (2019a), we find that additional entropy regularization component H makes 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 OOD (AdversarialQA) 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 (§A.1).

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

A.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 (§A.1).

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

B 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 $X = \text{SQuAD}_{\text{valid}}$ (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 bias significance 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*.

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.

C 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 5 and 6) 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 Bias significance.

Figure 5: 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.

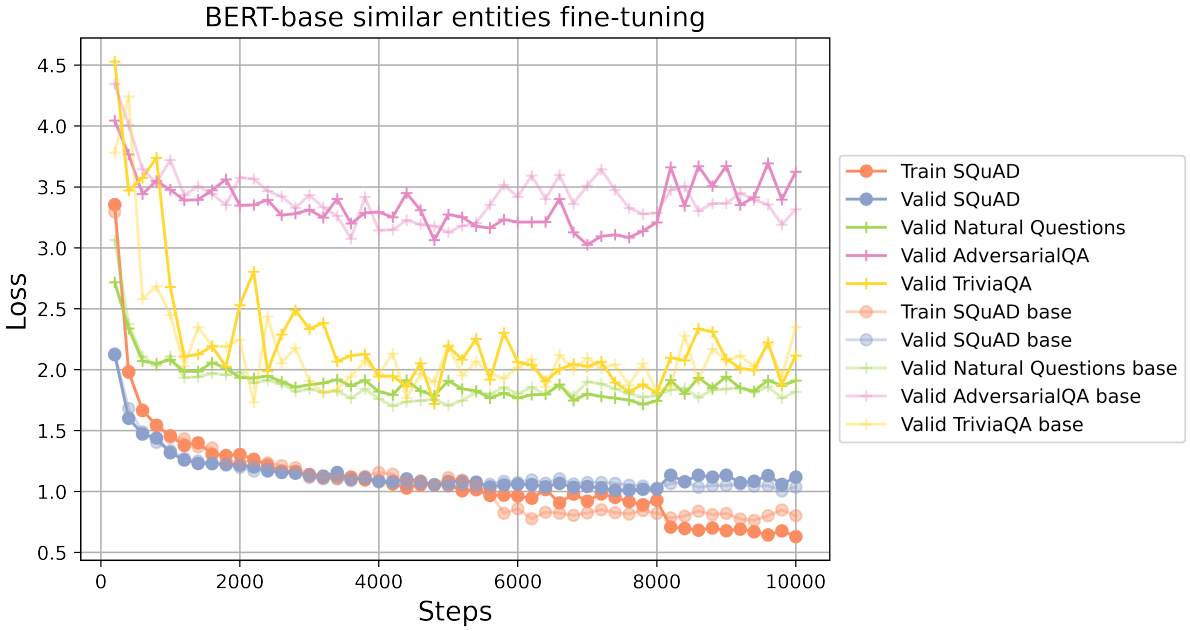

Figure 6: Development of validation loss of **ReSam** addressing ***sim-ents*** bias (darker plots) and standard fine-tuning (lighter plots) for Question Answering on SQuAD, also evaluated on other (OOD) datasets, for the first 10 000 steps.