Fighting Bias with Bias: Promoting Model Robustness by Amplifying Dataset Biases

Yuval Reif **Roy Schwartz**
School of Computer Science and Engineering, The Hebrew University of Jerusalem, Israel
{yuval.reif,roy.schwartz1}@mail.huji.ac.il

Abstract

NLP models often rely on superficial cues known as *dataset biases* to achieve impressive performance, and can fail on examples where these biases do not hold. Recent work sought to develop robust, unbiased models by filtering *biased* examples from training sets. In this work, we argue that such filtering can obscure the true capabilities of models to overcome biases, which might never be removed in full from the dataset. We suggest that in order to drive the development of models robust to subtle biases, dataset biases should be *amplified* in the training set. We introduce an evaluation framework defined by a *bias-amplified* training set and an *anti-biased* test set, both automatically extracted from existing datasets. Experiments across three notions of *bias*, four datasets and two models show that our framework is substantially more challenging for models than the original data splits, and even more challenging than hand-crafted challenge sets. Our evaluation framework can use any existing dataset, even those considered obsolete, to test model robustness. We hope our work will guide the development of robust models that do not rely on superficial biases and correlations. To this end, we publicly release our code and data.[1]

1 Introduction

NLP models often exploit repetitive patterns introduced during data collection, known as *dataset biases*, to achieve strong performance (Poliak et al., 2018; McCoy et al., 2019).[2] This trend has led to attempts of improving the evaluation of NLP models by creating test sets that are different from the training sets, e.g., from a different domain (Williams et al., 2018) or a different distribution (Koh et al., 2021), and challenge sets that focus on counterexamples to known biases in the training set, which we refer to as *anti-biased* examples (Jia and Liang, 2017; Naik et al., 2018; Utama et al., 2020).

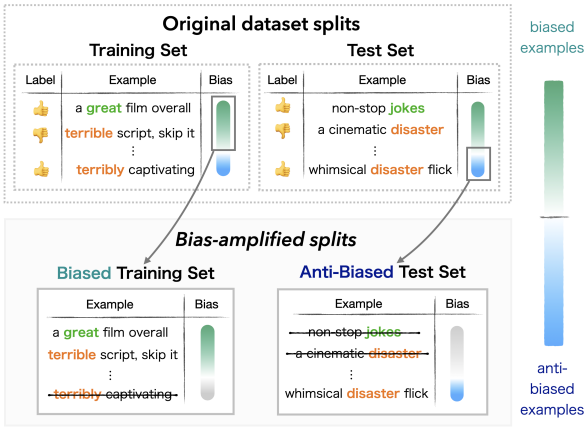

Figure 1: To guide the development of models robust to subtle biases, we propose to extract bias-amplified splits for existing benchmarks. Our approach first partitions a given dataset into *biased* and *anti-biased* instances. It then constructs a *biased* training set and an *anti-biased* test set, which are used to evaluate model generalization.

To address these gaps, some works used balancing techniques to create *unbiased* datasets, by filtering out *biased* examples (Zellers et al., 2018; Le Bras et al., 2020; Swayamdipta et al., 2020), or injecting *anti-biased* examples into the training sets (Nie et al., 2020; Liu et al., 2022a). In this work we argue that in order to encourage the development of robust models, we should in fact **amplify** biases in the training sets, while adopting the challenge set approach and making test sets *anti-biased* (Fig. 1).

Amplifying dataset biases might seem counter-intuitive at first. Our work follows recent work that challenged the assumption that biases can ever be fully removed from a given dataset (Schwartz and Stanovsky, 2022), arguing that models are able to pick up on very subtle phenomena even in partially balanced (or mostly unbiased) datasets (Gardner et al., 2021). As a result, dataset balancing, while potentially improving generalization, might make

[1] https://github.com/schwartz-lab-NLP/fight-bias-with-bias

[2] Instances that can be solved using such biases are typically referred to as "*biased*" (He et al., 2019).

Findings of the Association for Computational Linguistics: ACL 2023, pages 13169–13189

it harder to develop models that are resilient to such biases; these biases "hide" in the balanced training sets, and the way models handle them is hard to evaluate and make progress on.[3] Instead, we argue that academic benchmarks should include training splits that mainly consist of *biased* examples (see Fig. 2). Such splits will drive the development of robust models that generalize beyond biases, ideally even subtle ones.

We present a simple method to implement our approach (Sec. 2). Given a dataset in which both training and test sets are divided into *biased* and *anti-biased* subsets, we remove the *anti-biased* instances from the training set and the *biased* ones from the test set. The new splits then form a challenging evaluation setting. We assume that *biased* instances constitute the majority of a dataset (Gururangan et al., 2018; Utama et al., 2020), and thus the resulting training sets are similar in size to the original ones (though the test sets are smaller).

To discern *biased* and *anti-biased* instances, we consider three model-based approaches (Sec. 3): (a) dataset cartography (Swayamdipta et al., 2020), which uses training dynamics to profile the difficulty of learning individual data instances. In this approach, we identify instances that are hard-to-learn as *anti-biased* (Sanh et al., 2021; He et al., 2019); (b) partial-input models (Kaushik and Lipton, 2018; Poliak et al., 2018), which are forced to rely on bias, regarding instances on which they fail as *anti-biased*; and a method we introduce for identifying (c) *minority examples* (Tu et al., 2020; Sagawa et al., 2020), which groups a dataset's instances using deep clustering (Caron et al., 2018) and regards the minority-label instances within each cluster as *anti-biased*.

We apply our framework to *MultiNLI* (Williams et al., 2018) and *QQP* (Wang et al., 2018), on which trained models exceed human performance. We also experiment with two datasets that are considered more challenging: Adversarial NLI (*ANLI*; Nie et al., 2020) and WANLI (Liu et al., 2022b). We use a ROBERTA-BASE (Liu et al., 2019b) model for selecting *biased* and *anti-biased* instances according to each method, and evaluate the performance of ROBERTA and DEBERTA (He et al., 2021) LARGE models under our proposed setting (Sec. 4). While *anti-biased* instances are naturally challenging for models, amplifying biases in the training set makes them even more challenging; using the partial-input and minority examples methods, we observe mean absolute performance reductions of 15.8% and 31.8%, respectively. Using instances detected with dataset cartography leads to smaller (though still large) reductions of 10.1%.

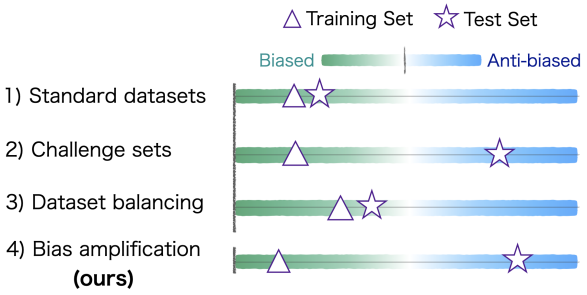

Figure 2: Different approaches to data collection. In standard datasets (1), the training and test sets mostly contain a majority of *biased* instances. Challenge sets (2) curate *anti-biased* test sets. Balancing and filtering methods (e.g., adversarial filtering, 3) collect *unbiased* training and test sets. Our framework (4) contains *biased* training sets and *anti-biased* test sets.

We compare *bias-amplified splits* to hand-crafted challenge sets such as *HANS* (McCoy et al., 2019), and find that our automatically-generated *anti-biased* test sets are both of similar difficulty to such challenge sets, and capture a more diverse set of biases. Our framework can further be used to augment existing challenge sets, as training on bias-amplified data increases their difficulty.

Next, we investigate how many *anti-biased* examples are required for generalization, by gradually re-inserting such instances to the training set (Liu et al., 2019a). While models greatly benefit from observing small amounts of *anti-biased* instances, *anti-biased* test sets remain challenging, and additional performance gains require much larger quantities (Sec. 5). We then show that standard debiasing methods applied to bias-amplified training sets lead to little to no gains in performance (Sec. 6).

Our findings may change the way we evaluate the robustness of NLP models, and in particular their level of generalization beyond the biases of their training sets. Our method requires no new annotation or any task-specific expertise. It allows to rejuvenate datasets previously considered as obsolete, and thus reuse the intensive efforts used in their curation. We release our new dataset splits along with code for automatically creating bias-amplified splits for other datasets.

[3] Indeed, training on adversarial data doesn't necessarily generalize to non-adversarial data (Kaushik et al., 2021).

2 Amplifying Dataset Biases to Advance Model Robustness

This section motivates our approach in view of recent developments in NLP, provides a general overview of the framework we use to implement it, and discusses its applications.

2.1 Motivation: Data Balancing Hides Biases

This paper focuses on the problem of creating robust models that generalize beyond dataset biases. A common approach to addressing this problem is removing these biases from the training data (Zellers et al., 2018; Le Bras et al., 2020). This approach is intuitive—if a model doesn't observe these biases in the first place, it is less likely to learn them, and will thus generalize better.

Despite the appeal of this approach, it suffers from several problems. First, recent work has argued that models are sensitive to very fine-grained biases, which are hard to detect and filter (Gardner et al., 2021). Other works have shown that training on bias-filtered datasets does not necessarily lead to better generalization (Kaushik et al., 2021; Parrish et al., 2021), indicating that while such training sets are less biased, models might still rely on biases to solve them. Finally, recent studies argued that even with our utmost efforts, we may never be able to create datasets that contain no exploitable biases (Linzen, 2020; Schwartz and Stanovsky, 2022).

As a result, this paper argues that mitigating the negative effect of dataset biases is not only a data problem, but needs to also come from better *modeling*. But how can we create a testbed for developing models that overcome these biases? We argue that training on datasets filtered for such biases will not suffice in developing such models, and in fact make it harder to do so; as subtle biases still "hide" inside filtered training sets, it is much harder to track them, evaluate their impact and importantly—develop models that learn to ignore them.

Instead, in this paper we propose that when evaluating model robustness, dataset biases should be **amplified** by training mostly on *biased* instances, while using *anti-biased* instances for evaluation (Fig. 2). This simple setting defines a challenging test, where models must counteract dataset biases and learn generalizable solutions in order to succeed, as the *anti-biased* test set cannot be solved using the *biased* training set's statistical cues.

2.2 Framework for Amplifying Dataset Biases

We describe our approach for amplifying dataset biases during training to evaluate model generalization. Given a dataset split into training and test sets $\mathcal{D} = \mathcal{D}^{\text{train}} \cup \mathcal{D}^{\text{test}}$, we begin by dividing its instances across both splits into *biased* and *anti-biased* subsets.[4] To evaluate a model's robustness, we first train it on the portion of *biased* train instances $\mathcal{D}^{\text{train}}_{\textit{biased}}$. We assume most data instances are *biased* (Gururangan et al., 2018), so this process results in small reductions in training set sizes compared to $\mathcal{D}^{\text{train}}$. We then evaluate the model on the *anti-biased* test instances $\mathcal{D}^{\text{test}}_{\textit{anti-biased}}$, and compare it to the performance of the same model trained on the full training set. Drops in performance between the two indicate that the model struggles to overcome its training set biases.

2.3 Discussion

Applications We suggest our framework as a tool for studying and evaluating models. As such, it is orthogonal to data collection procedures. Importantly, we do not suggest to intentionally collect biased data when curating new datasets. Nonetheless, data collected in large quantities tends to contain unintended regularities (Gururangan et al., 2018). We therefore propose to use bias-amplified splits to complement benchmarks with challenging evaluation settings that test model robustness, in addition to the dataset's main training and test sets.

Such splits, when created using the methods we consider in this work, can be created automatically and efficiently for any dataset. These include newly collected datasets, but also existing ones, such as obsolete benchmarks on which model performance is too high to measure further progress, allowing for the rejuvenation and reuse of benchmarks.

***Anti-biased* vs. challenge sets** Our framework provides an evaluation environment to assess model robustness, similar to challenge sets. However, unlike challenge sets, which are often manually curated with protocols designed to create difficult examples, our approach is automatic and uses data collected using the exact same protocol as the model's training data. Still, we find that *anti-biased* test sets are challenging for models and can capture more diverse biases, and moreover—that training on *bias-amplified* data further enhances

[4] We consider three different notions of *biased* instances (Sec. 3), but other definitions (e.g., Godbole and Jia, 2023, see Sec. 7 for discussion) could be integrated into our framework.

the difficulty of existing challenge sets (Sec. 4.2). Consequently, our framework can be employed to evaluate robustness in tasks where challenge sets are unavailable, or in conjunction with existing challenge sets for a more comprehensive evaluation.

Can models generalize from *biased* data? A natural question to ask about our approach is whether we can truly expect models to generalize from a biased training distribution. Although the *biased* training sets could be solved by capturing only a subset of relevant features, their instances can still provide valuable information for learning additional features that are important for generalization yet under-utilized by models (Shah et al., 2020; Geirhos et al., 2020). Previous work has proposed techniques to encourage models to learn diverse, unbiased representations from extremely biased training distributions, mostly focusing on domains outside of NLP (Kim et al., 2019; Bahng et al., 2020; Pezeshki et al., 2021). This is likely due to the difficulty of defining and controlling biased distributions in textual domains. Our work paves the way for implementing and evaluating such methods specifically for NLP.

Related to this concern is our decision to leave **no** *anti-biased* instances in the training set. Indeed, it is likely that for many biases, at least *some* counter-examples will be found in the training set. We admit that this decision is not a major component of our approach, and it could be easily implemented with a *small* number of *anti-biased* instances in the training set instead. To avoid deciding on the numeric definition of small, and to make the setup as challenging as possible, we experiment throughout this paper with no (identified) *anti-biased* instances in training. In Sec. 5 we study the effect of using limited amounts of such counter-examples, by reinserting some *anti-biased* instances into training.

3 Definitions of *Biased* and *Anti-biased* Examples

Our approach requires a drop-in method for classifying a dataset's examples into *biased* and *anti-biased* instances. We consider the following model-based methods for doing so. We note that none of them requires any prior knowledge or task-specific expertise. All methods can be computed automatically at the reasonable cost of training and evaluating a (possibly smaller) model on the dataset.

Dataset Cartography (Swayamdipta et al., 2020) is a method to automatically characterize a dataset's instances according to their contribution to a model's performance, by tracking a model's training dynamics. Specifically, measuring each instance's **confidence**—the mean of the predicted model probabilities for the gold label across training epochs—reveals a region of *easy-to-learn* instances with high confidence which the model consistently predicts correctly throughout training, and a region of low-confidence *hard-to-learn* instances, on which the model consistently fails during training. We follow previous work which considered instances that models find easy or hard to solve as more likely to be *biased* or *anti-biased*, respectively (Sanh et al., 2021; He et al., 2019).

To estimate the confidence of test instances, we make predictions with a partially trained model at the end of each epoch on the test set (as typically done on the validation set), and use the average confidence scores across epochs.[5] To choose *anti-biased* examples, we use the $q\%$ most *hard-to-learn* instances in each of the training and the test sets individually, where q is a hyperparameter. We consider all other examples as *biased*.

Partial-input baselines is a common method for identifying annotation artifacts in a dataset. The method works by examining the performance of models that are restricted to using only part of the input. Such models, if successful, are bound to rely on unintended or spurious patterns in the dataset. Examples include question-only models for visual question answering (Goyal et al., 2017), ending-only models for story completion (Schwartz et al., 2017) and hypothesis-only models for natural language inference (Poliak et al., 2018).

Held-out instances where such baselines fail are considered *anti-biased* and less likely to contain artifacts (Gururangan et al., 2018).[6] Generating a *biased* training set for this method is not trivial, as the partial-input model is likely to fit to the training data during training, and thus almost all examples will be labeled *biased*. We therefore follow the dataset cartography approach with a partial-input baseline, and compute the mean confidence score for each instance across epochs. We select the $q\%$

[5] We emphasize that we are **not** fine-tuning on the test set, nor using it to select any hyperparameters. This process only annotates the *biased* and *anti-biased* portions of the test set.

[6] Such instances might still contain more complex artifacts that are only detectable when jointly inspecting all parts of the input (Feng et al., 2019).

most hard-to-learn instances as *anti-biased*.

Minority examples Current models are typically sensitive to *minority examples* that defy common statistical patterns found in the rest of the data, especially when the amount of such examples in the training set is scarce (Tu et al., 2020; Sagawa et al., 2020). Minority examples are often detected by heuristically searching for spurious features correlated with one label in the instances of another label (e.g., high word overlap between two non-paraphrase texts). Motivated by recent work that leverages instance similarity in the representation space of fine-tuned language models for various use cases (Liu et al., 2022b; Pezeshkpour et al., 2022), we propose a model-based clustering approach to automatically detect minority examples.

We follow a three-step approach. First, we cluster the *training* set using [CLS] token representations extracted from a model trained on the dataset. Second, to detect minority examples in the training set, we inspect the distribution of instances over the task labels L within each cluster c_i. We define a cluster c_i's *majority label* as the label $\ell_i \in L$ associated with the most instances in c_i. We consider all other labels as c_i's minority labels. Instances belonging to their cluster's minority labels are regarded as *minority examples*, and accordingly *anti-biased*, while and all others are considered *biased*. Finally, to detect minority examples in the *test* set, we extract [CLS] representations for all test instances, and assign each instance to the cluster of its nearest neighbor in the *training* set using Euclidean distance. If the test instance (x, y) is assigned to cluster c_i, we consider (x, y) as a majority example iff it belongs to c_i's majority label, i.e., if $y == \ell_i$. [7]

Our preliminary experiments show that standard clustering algorithms tend to create label-homogeneous clusters, i.e., they are less likely to cluster together instances from different labels. We thus use DEEPCLUSTER (Caron et al., 2018), which we find to create more label-diverse clusters. DEEPCLUSTER alternates between grouping a model's representations with a standard clustering algorithm[8] to produce pseudo-labels, and fine-tuning a new pretrained model to predict these pseudo-labels. We perform one iteration of deep clustering and then cluster the representations of the DEEPCLUSTER model to obtain the final clustering. App. C shows details and preliminary results on alternative clustering methods.

4 Models Struggle with Amplified Biases

We next use our framework to evaluate the extent to which models generalize beyond the biases of their training sets.

4.1 Experimental Setup

We create bias-amplified splits for four datasets: two (*QQP*, Wang et al., 2018; and *MultiNLI*, Williams et al., 2018) that were shown to contain considerable biases (Zhang et al., 2019; Gururangan et al., 2018); and two additional datasets (*ANLI*, Nie et al., 2020; and WANLI, Liu et al., 2022b) designed to contain smaller proportions of *biased* instances. QQP is a duplicate question identification dataset, while the other three are natural language inference (NLI) datasets.

We split all datasets into *biased* and *anti-biased* parts according to each of the three methods described in Sec. 3. We use a ROBERTA-BASE (Liu et al., 2019b) model for all three methods: we fine-tune the model on each dataset to compute training dynamics for *dataset cartography*, and also to extract and cluster [CLS] representations for identifying *minority examples*; we separately train the model on partial inputs to obtain training dynamics for *partial-input baselines*. We use hypothesis-only baselines for NLI datasets. For QQP, we use the first question of each pair.

We then evaluate the performance of ROBERTA and DEBERTA (He et al., 2021) LARGE models under our proposed framework. We train models on the *biased* training split obtained from each of the three methods, and report their performance on the corresponding *anti-biased* test sets.[9] Since the number of *biased* training instances is induced by the clustering in the *minority examples* approach, but is a hyperparameter q for the two other approaches, we adjust q to create equally sized training sets for all three methods. This results in 79% of the training set for *MultiNLI*, 82% for *QQP* and

[7] Note that unlike the methods described above for detecting *anti-biased* subsets, the minority examples approach does not require a pre-determined size for the resulting subset, as it is induced by the clustering.

[8] We use Ward's method (Ward Jr, 1963), a popular deterministic algorithm for hierarchical clustering which has the same objective function as K-means.

[9] We use the validation sets of *QQP* and WANLI, and the validation-matched set for *MultiNLI*, as their test sets are not publicly available.

Train		*MultiNLI*				*QQP*				WANLI				*ANLI*			
		Orig.	Cart.	ParIn	Mino.	Orig.	Cart.	ParIn	Mino.	Orig.	Cart.	ParIn	Mino.	Orig.	Cart.	ParIn	Mino.
	full	$90.4_{0.2}$	$59.9_{0.7}$	$79.7_{0.6}$	$71.9_{0.3}$	$92.0_{0.1}$	$59.4_{0.4}$	$78.6_{0.2}$	$73.5_{0.3}$	$76.2_{0.1}$	$19.7_{3.4}$	$59.5_{0.6}$	$60.2_{1.6}$	$55.4_{0.1}$	$14.8_{0.6}$	$34.9_{1.0}$	$44.3_{0.4}$
	rand	$90.3_{0.1}$	$59.5_{0.7}$	$79.7_{0.7}$	$71.9_{1.0}$	$91.6_{0.1}$	$57.8_{0.4}$	$78.0_{0.5}$	$71.8_{1.0}$	$76.0_{0.1}$	$17.6_{0.9}$	$58.0_{3.0}$	$59.2_{2.2}$	$55.1_{0.7}$	$15.7_{1.0}$	$34.1_{1.4}$	$44.2_{1.1}$
	bias	$88.4^{*}_{0.7}$	$51.7_{0.5}$	$68.2_{0.3}$	$50.5_{1.2}$	$88.3^{*}_{1.9}$	$49.0_{0.3}$	$60.3_{1.8}$	$31.3_{0.4}$	$74.9^{*}_{1.0}$	$13.7_{0.8}$	$43.5_{2.9}$	$25.8_{2.4}$	$51.2^{*}_{1.9}$	$5.8_{0.7}$	$16.0_{1.0}$	$12.3_{0.8}$

Table 1: Accuracy of our approach with RoBERTa-LARGE models. Different rows correspond to different training schemes: the full dataset (*full*), a biased subset (*bias*) and a random subset the size of *bias* (*rand*). Column groups correspond to different datasets. Individual columns represent testing schemes: the original validation/test set (Orig.) and the *anti-biased* test splits: *dataset cartography* (Cart.), *partial-input* (ParIn) and *minority examples* (Mino.). Reported values are averaged across three random seeds, with standard deviation as subscripts. Results in the last row (*bias*) are of training on the *biased* split and testing on the respective *anti-biased* split, except for Orig. values (marked with $*$), which are averaged over runs on all three *biased* splits. Model evaluation on bias-amplified splits results in weak performance on *anti-biased* test instances compared to the original data splits.

ANLI, and 87% for WANLI.[10] See App. A.2 for more details on the experimental setup.

Baselines We compare against two baselines: the original training split (**100% train**) and a random sample of the same size as the *biased* training splits (**random**). In addition to the *anti-biased* test set, we also report performance on the original test set to validate that the model's training data (the *biased* training instances) is sufficient for learning the task.

Hyperparameters selection Our approach for identifying *minority examples* is based on clustering the representations of a fine-tuned model. The clustering algorithm we use, DeepCluster (Sec. 3), has three hyperparameters: the number of final clusters k, the number of pseudo-labels m for representation learning, and the Transformer layer from which [CLS] representations are extracted for clustering. We use $k = 10$ clusters for all datasets, and search for a good configuration for the other two hyperparameters on *SST-2* (Socher et al., 2013): for each set of hyperparameters, we apply the *minority examples* method to create *biased* training and *anti-biased* test splits, and train two RoBERTa-BASE models—one on the *biased* training split, and a baseline model on an equally-sized random training subset. We select the hyperparameters that lead to the largest performance drop on *anti-biased* test instances between the two, and use them in all further experiments to cluster other datasets; see App. C.3 for details.

[10]When selecting *minority examples* for *MultiNLI* and *QQP*, we consider all labels but a cluster's *majority label* as minority labels. Using this setting for *ANLI* and WANLI results in specifying more than 40% of the training set as *minority examples*. This leaves too few *biased* instances for training and substantially changes the original training distribution. Therefore, for these datasets, we use the label with the *least* instances within a cluster as its minority-label.

4.2 Results

Models struggle with *biased* training sets Tab. 1 shows our results for RoBERTa-LARGE. We observe that the baseline models struggle with all *anti-biased* test sets, even when training on the full training set. The *anti-biased* test splits based on *dataset cartography* prove to be the most initially difficult, with the splits created using the two other methods overall similar in difficulty. Still, model performance on *anti-biased* instances drops further when training on *biased* training splits; taking the mean across datasets, performance drops by 8.4% for *dataset cartography*-based splits, 16.2% for *partial-input*, and 32.5% for *minority examples*. Results for DeBERTa-LARGE (App. B.1) follow the same trends, with mean performance reductions of 11.8% for *dataset cartography*, 15.4% for *partial-input*, and 31.1% for *minority examples*.

We also observe that training on *biased* splits leads to minor reductions on the full test sets, indicating that while current models trained on our training splits fail to generalize beyond the biases in these sets, they are seemingly able to learn the tasks at hand.

***Anti-biased* test sets are as challenging as manual challenge sets** We further compare model performance on our *anti-biased* test splits to performance on challenge sets collected manually. Particularly, we compare the splits created with the *minority examples* method for *MultiNLI* and *QQP*, to the *HANS* (McCoy et al., 2019) and *PAWS* (Zhang et al., 2019) challenge sets, respectively.

Our results (Tab. 2 for *HANS*, Tab. 3 for *PAWS*) show that, when training on the full dataset, our automatically curated test splits are more difficult than the *HANS* challenge set, but not as challenging as *PAWS* (Mean column). Interestingly, train-

	MultiNLI-anti-biased				*HANS*		
	E	N	C	Mean	E	¬E	Mean
full	$51.4_{3.5}$	$75.5_{1.0}$	$77.8_{2.3}$	$71.9_{0.3}$	$99.8_{0.2}$	$56.5_{0.8}$	$78.2_{0.5}$
rand	$51.1_{3.2}$	$75.6_{2.8}$	$78.0_{1.6}$	$71.9_{1.0}$	$99.8_{0.1}$	$55.7_{1.1}$	$77.8_{0.5}$
biased	$45.1_{1.7}$	$48.9_{2.3}$	$57.6_{1.4}$	$50.5_{1.2}$	$99.8_{0.1}$	$2.9_{0.9}$	$51.4_{0.4}$

Table 2: Accuracy of RoBERTa-LARGE models trained on different subsets of *MultiNLI*, when evaluated on the dataset's *anti-biased* test split and on *HANS*. Model performance is reported per label (Entailment: *E*, Neutral: *N*, Contradiction: *C*, Not entailment: ¬*E*) and over all examples (*Mean*). *Biased* and *anti-biased* splits are created with the *minority examples* method. *Anti-biased* sets are comparably difficult to manually designed challenge sets, yet capture diverse biases in *all* task labels.

	QQP-anti-biased			*PAWS*		
	D	¬ D	Mean	D	¬ D	Mean
full	$80.9_{0.4}$	$69.9_{0.5}$	$73.5_{0.3}$	$94.2_{0.6}$	$17.7_{3.5}$	$51.5_{1.7}$
rand	$80.0_{0.4}$	$67.8_{1.3}$	$71.8_{1.0}$	$95.2_{0.3}$	$13.4_{1.5}$	$49.6_{0.9}$
biased	$27.9_{2.3}$	$33.0_{1.4}$	$31.3_{0.4}$	$95.6_{0.8}$	$4.9_{0.8}$	$44.5_{0.1}$

Table 3: Accuracy of RoBERTa-LARGE models trained on different subsets of *QQP*, when evaluated on the dataset's *anti-biased* test split and on *PAWS*. Model performance is reported per label (Duplicate: *D*, Not duplicate: ¬*D*) and over all examples (*Mean*). *Biased* and *anti-biased* splits are created with the *minority examples* method.

ing on *biased* splits (final row) makes the challenge sets dramatically more difficult, but our *anti-biased* splits are even more challenging in this setup—the model performs 0.9% worse on *MultiNLI* compared to *HANS*, and 13.2% worse on *QQP* compared to *PAWS*.

We further find that *anti-biased* test splits are more *diverse* than the challenge sets, as difficult instances affected by biases arise in *all* labels in the *anti-biased* splits, while mostly in one label in the challenge sets. Our results suggest that bias-amplified splits can augment existing challenge sets by boosting their difficulty or uncovering instances that influence the biases they test.

Discussion Overall, bias-amplified splits prove to be extremely difficult for strong models. Such splits could be used to identify models that successfully generalize beyond substantial biases, and are more likely to overcome subtler ones. Importantly, bias amplification remains challenging even when applied to recent datasets that contain fewer *biased* instances (e.g., *ANLI* and WANLI), or when compared to hand-crafted challenge sets. They could therefore be used to complement model evaluation on future, more challenging datasets. Finally, our

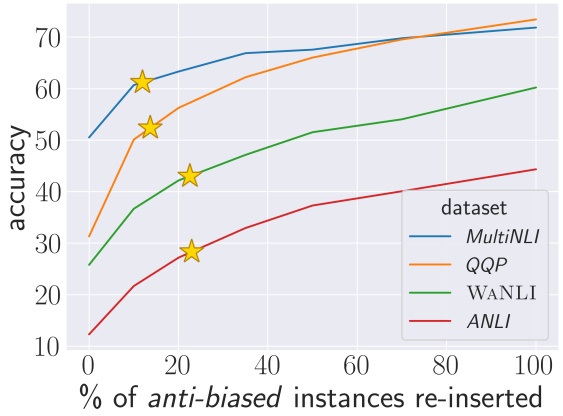

Figure 3: Accuracy for RoBERTa-LARGE models fine-tuned on bias-amplified splits created with the *minority examples* method, while gradually reinserting *anti-biased* instances back into the training set. Reported values are averaged across three random seeds. We interpolate and place stars (★) at points where the model regains 50% of its original performance. Models generalize from small amounts of *anti-biased* instances, but require much larger quantities to achieve comparable performance gains.

splits can be created automatically for any existing dataset, even those for which model performance on the standard splits exceeds human performance, such as *MultiNLI* and *QQP*.

5 How Many *Anti-biased* Examples are Needed for Generalization?

So far, we have seen that amplifying dataset biases by eliminating **all** *anti-biased* instances from the training set uncovers shortcomings in model generalization. We next study the effect of allowing **some** *anti-biased* instances in the training set (Liu et al., 2019a). We fine-tune RoBERTa-LARGE on all four datasets using the *biased* splits created using the *minority examples* method, while gradually reinserting 10%, 20%, 35%, 50% and 70% of the *anti-biased* instances back into the training set.[11]

Our results (Fig. 3) show that reinserting 20% of the *anti-biased* training instances allows the model to close approximately 50% of the gap from its baseline performance on the *anti-biased* test set. Surprisingly, performance grows slowly when restoring additional *anti-biased* instances, and does not match the full training set's levels even when adding 70% of *anti-biased* instances. This indi-

[11]Note that the *anti-biased* instances still constitute a minority within each cluster, as even 100% of the *anti-biased* instances is considered a minority.

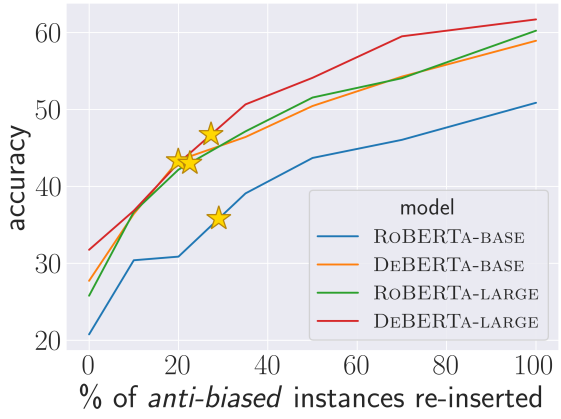

Figure 4: Accuracy for models trained on bias-amplified splits of WANLI created with the *minority examples* method, while gradually reinserting *anti-biased* instances back into the training set.

cates that the model is capable of generalizing from small amounts of *anti-biased* instances, but is inefficient in gaining further improvements. Results for the other models (Fig. 4) show a similar trend.

On the one hand, our results encourage careful data collection in order to fill gaps in dataset coverage (Parrish et al., 2021; Liu et al., 2019a). On the other hand, our findings indicate that data curation is not a sufficient solution, as models struggle on minority examples even when observing all available instances, and collecting more instances results in smaller further gains. Thus, it is also necessary to develop robust models that can better generalize from biased data. Our proposed framework provides a testbed for doing so.

6 The Effect of Debiasing Methods

Recently proposed methods were shown to be effective in improving the out-of-distribution generalization of models, either by adjusting the training loss to account for biased instances (**model debiasing**; He et al. 2019; Clark et al. 2019), or by filtering the training set to increase the proportions of different kinds of instances found to be advantageous for generalization (**data filtering**; Le Bras et al. 2020; Yaghoobzadeh et al. 2021; Liu et al. 2021). We now examine whether such methods improve the generalization of models trained on bias-amplified training sets to *anti-biased* test instances.

We consider a RoBERTa-large model trained on bias-amplified splits of *MultiNLI* and *QQP* based on *minority examples*. For model debiasing, we apply the self-debiasing framework suggested

	MultiNLI-anti-biased	*QQP-anti-biased*
biased	$50.5_{1.2}$	$31.3_{0.4}$
self-debiasing	$50.7_{0.9}$	$33.1_{1.7}$
ambiguous filtering	$51.4_{0.4}$	$32.5_{1.9}$
100% train	$71.9_{0.3}$	$73.5_{0.3}$

Table 4: Accuracy of RoBERTa-large models trained on *MultiNLI* and *QQP* with different training schemes: a *biased* subset, two debiasing methods applied to the *biased* subset, and the full training set. We use the *biased* and *anti-biased* splits created with the *minority examples* method. Applying model debiasing or data filtering approaches in the bias-amplified setting results in only slight improvements on the *anti-biased* test sets.

by Utama et al. (2020)[12] with example reweighting (Schuster et al., 2019) to down-weight the loss function for biased instances; for data filtering, we apply dataset cartography to identify ambiguous instances—examples for which the model's confidence in the gold label exhibits high variability across training epochs—and train on the 33% most ambiguous ones, as shown to benefit generalization in Swayamdipta et al. (2020). Importantly, we apply both methods to the bias-amplified training split (rather than the original training set) and do not train on any other instances during the debiasing or filtering procedures.

Our results (Tab. 4) show that neither debiasing nor filtering result in substantial improvements on *anti-biased* data. This indicates that such methods are less effective when training sets lack sufficient *anti-biased* instances, and highlights the need for methods that could improve model generalization when additional data curation is impractical. Our findings are also in line with recent results showing that various robustness interventions struggle with improving upon standard training in real-world distribution shifts (Koh et al., 2021) or dataset shifts (Taori et al., 2020; Awadalla et al., 2022).

7 Related Work

Biased splits The concept of re-organizing a dataset's training and test splits is often used to create more challenging evaluation benchmarks from existing datasets by inserting bias into the training set. Søgaard et al. (2021) showed that using biased splits better approximates real-world perfor-

[12] In self-debiasing, a biased version of the model is used to detect biases. We follow Utama et al. (2020) and obtain such models by training on 2000 examples and 3 epochs for *MultiNLI*, and 500 examples and 4 epochs for *QQP*.

mance compared to standard, random splits. Koh et al. (2021) and Santurkar et al. (2021) simulated real-world distribution shifts by filtering out different kinds of data from the training and test sets, based on manually crafted heuristics. Agrawal et al. (2018) ignored the dataset's original training and test splits altogether and re-split instances to create biased splits for VQA using dataset-specific heuristics. Unlike such approaches, our method automatically constructs biased splits using dataset-agnostic approaches, and follows the original training and test splits. Concurrently to this work, Godbole and Jia (2023) re-split datasets by placing all examples that are assigned lower likelihood by an LM in the test set, and more likely examples in the training set. In some sense, that work also creates an "easy" training set and a "hard" test set, and can thus be considered a special case of our approach.

Challenge sets Given the exceptional performance of modern NLP tools on standard benchmarks, challenging test sets were created to better assess model capabilities across various tasks (Isabelle et al., 2017; Naik et al., 2018; Marvin and Linzen, 2018). Such approaches often rely on human experts to identify model weaknesses and create challenging test cases using instance perturbations (Jia and Liang, 2017; Glockner et al., 2018; Belinkov and Bisk, 2018; Gardner et al., 2020) or rule-based data creation protocols (McCoy et al., 2019; Jeretic et al., 2020). Some approaches automated certain parts of these procedures, yet still require human design or annotation (Bitton et al., 2021; Li et al., 2020; Rosenman et al., 2020).

Inserting instances from challenge sets to the training set was shown to potentially alleviate their difficulty (Liu et al., 2019a), perhaps similarly to how model performance in our framework improves when reintroducing *anti-biased* examples to the training set (Sec. 5). Other work extracted challenging test subsets from existing benchmarks for focused model evaluation (Gururangan et al., 2018). Our framework can similarly be used to better evaluate model generalization, but without requiring additional annotations or task-specific expertise, and using data that was collected in the exact same procedure as the model's training data. We further showed (Sec. 4) that our framework can be used along with existing challenge sets to increase their difficulty.

Dataset balancing Recent work proposed methods to collect benchmarks with balanced and ideally unbiased training and test splits. Such benchmarks often use a model-in-the-loop during data collection and task crowd workers to write examples on which models fail (Bartolo et al., 2020; Nie et al., 2020; Kiela et al., 2021; Talmor et al., 2021), or used adversarial filtering to remove examples from existing or newly collected datasets that were easily solved by models (Zellers et al., 2018, 2019; Dua et al., 2019; Le Bras et al., 2020; Sakaguchi et al., 2021). Parrish et al. (2021) proposed to use an expert linguist-in-the-loop during crowdsourcing to improve data quality and diversity. Other work used generative methods to enrich existing datasets and compose new machine-generated examples similar to challenging seed examples (Lee et al., 2021; Liu et al., 2022a). Other studies argued that despite our best efforts, we may never be able to create datasets that are truly balanced (Linzen, 2020; Schwartz and Stanovsky, 2022). Our framework can be used to expose biases in such datasets and to automatically augment them with more challenging evaluation splits.

8 Conclusion

Recent approaches in NLP attempted to eliminate dataset biases from training sets to produce robust models and reliable evaluation settings, yet model generalization remains a challenge, and subtler biases persist. In this work, we argued that to promote robust modeling, models should instead be evaluated on datasets with *amplified* biases, such that only true generalization will result in high performance. We presented a simple framework to automatically create bias-amplified splits for a given dataset, finding that such splits are difficult for strong models when created for either obsolete or difficult datasets, and could potentially expose differences in generalization capabilities between models. Our results indicate that bias amplification could ease the creation of robustness evaluation tests for new datasets, as well as inform the development of robust methods.

Acknowledgments

We thank Inbal Magar, Gal Patel, Gabriel Stanovsky and Hila Gonen for their insightful comments that contributed to this paper. We also thank our anonymous reviewers for their constructive feedback. This work was supported in part by the

Israel Science Foundation (grant no. 2045/21).

Limitations

In our experiments, we evaluated models by fine-tuning on bias-amplified splits, but we did not explore the robustness of few-shot methods. Such methods are intuitively less likely to be affected by slight changes in the distribution of examples they observe. However, recent work has shown that they could still be affected by dataset biases (Utama et al., 2021; Li et al., 2022), and we will use our framework to explore this in future work.

We note that our approach is less suitable for datasets with relatively small test sets. In such cases, extracting an *anti-biased* test split, which consisted of 13-21% of the original test set in the benchmarks we considered, will result in a test set too small to reliably evaluate models. However, the methods we used to extract bias-amplified splits (Sec. 3) could be tuned to produce larger test sets (while keeping the amount of *anti-biased* instances in the training set relatively small), e.g., by selecting a lower number of *biased* training instances (q, Sec. 4.1).

Throughout this paper, we used the term "bias" to describe statistical regularities in datasets that can be exploited by models as unintended shortcut solutions. While we do not explore model robustness to other types of data biases (e.g., different kinds of societal biases) our framework could potentially be used to evaluate how models handle such cases by revising the definitions of *biased* and *anti-biased* instances used to create the evaluation splits. We leave such applications of our framework to future work.

References

Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. 2018. Don't just assume; look and answer: Overcoming priors for visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4971–4980.

Anas Awadalla, Mitchell Wortsman, Gabriel Ilharco, Sewon Min, Ian Magnusson, Hannaneh Hajishirzi, and Ludwig Schmidt. 2022. Exploring the landscape of distributional robustness for question answering models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5971–5987, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Hyojin Bahng, Sanghyuk Chun, Sangdoo Yun, Jaegul Choo, and Seong Joon Oh. 2020. Learning de-biased representations with biased representations. In *International Conference on Machine Learning*, pages 528–539. PMLR.

Max Bartolo, Alastair 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.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *International Conference on Learning Representations*.

Yonatan Bitton, Gabriel Stanovsky, Roy Schwartz, and Michael Elhadad. 2021. Automatic generation of contrast sets from scene graphs: Probing the compositional consistency of GQA. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 94–105, Online. Association for Computational Linguistics.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. 2018. Deep clustering for unsupervised learning of visual features. In *European Conference on Computer Vision*.

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

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

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2368–2378, Minneapolis, Minnesota. Association for Computational Linguistics.

Shi Feng, Eric Wallace, and Jordan Boyd-Graber. 2019. Misleading failures of partial-input baselines. In

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 5533–5538, Florence, Italy. Association for Computational Linguistics.

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. 2020. Evaluating models' local decision boundaries via contrast sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1307–1323, Online. Association for Computational Linguistics.

Matt Gardner, William Merrill, Jesse Dodge, Matthew Peters, Alexis Ross, Sameer Singh, and Noah A. Smith. 2021. Competency problems: On finding and removing artifacts in language data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1801–1813, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

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

Ameya Godbole and Robin Jia. 2023. Benchmarking long-tail generalization with likelihood splits. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 963–983, Dubrovnik, Croatia. Association for Computational Linguistics.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913.

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

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

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTa: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Pierre Isabelle, Colin Cherry, and George Foster. 2017. A challenge set approach to evaluating machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2486–2496, Copenhagen, Denmark. Association for Computational Linguistics.

Paloma Jeretic, Alex Warstadt, Suvrat Bhooshan, and Adina Williams. 2020. Are natural language inference models IMPPRESsive? Learning IMPlicature and PRESupposition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8690–8705, Online. Association for Computational Linguistics.

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

Divyansh Kaushik, 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.

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

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. 2021. Dynabench: Rethinking benchmarking in NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4110–4124, Online. Association for Computational Linguistics.

Byungju Kim, Hyunwoo Kim, Kyungsu Kim, Sungjin Kim, and Junmo Kim. 2019. Learning not to learn: Training deep neural networks with biased data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9012–9020.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton Earnshaw, Imran Haque, Sara M Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. 2021. Wilds: A benchmark of in-the-wild distribution shifts. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5637–5664. PMLR.

Ronan Le Bras, Swabha Swayamdipta, Chandra Bhagavatula, Rowan Zellers, Matthew Peters, Ashish Sabharwal, and Yejin Choi. 2020. Adversarial filters of dataset biases. In *International Conference on Machine Learning*, pages 1078–1088. PMLR.

Kenton Lee, Kelvin Guu, Luheng He, Tim Dozat, and Hyung Won Chung. 2021. Neural data augmentation via example extrapolation. *arXiv preprint arXiv:2102.01335*.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Chuanrong Li, Lin Shengshuo, Zeyu Liu, Xinyi Wu, Xuhui Zhou, and Shane Steinert-Threlkeld. 2020. Linguistically-informed transformations (LIT): A method for automatically generating contrast sets. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 126–135, Online. Association for Computational Linguistics.

Xiang Lorraine Li, Adhiguna Kuncoro, Jordan Hoffmann, Cyprien de Masson d'Autume, Phil Blunsom, and Aida Nematzadeh. 2022. A systematic investigation of commonsense knowledge in large language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11838–11855, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Tal Linzen. 2020. How can we accelerate progress towards human-like linguistic generalization? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5210–5217, Online. Association for Computational Linguistics.

Alisa Liu, Swabha Swayamdipta, Noah A. Smith, and Yejin Choi. 2022a. WANLI: Worker and AI collaboration for natural language inference dataset creation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6826–6847, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Alisa Liu, Swabha Swayamdipta, Noah A. Smith, and Yejin Choi. 2022b. Wanli: Worker and ai collaboration for natural language inference dataset creation.

Evan Z Liu, Behzad Haghgoo, Annie S Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. 2021. Just train twice: Improving group robustness without training group information. In *International Conference on Machine Learning*, pages 6781–6792. PMLR.

Nelson F. Liu, Roy Schwartz, and Noah A. Smith. 2019a. Inoculation by fine-tuning: A method for analyzing challenge datasets. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2171–2179, Minneapolis, Minnesota. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar S. Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke S. Zettlemoyer, and Veselin Stoyanov. 2019b. RoBERTa: A robustly optimized BERT pretraining approach. ArXiv:1907.11692.

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Brussels, Belgium. Association for Computational Linguistics.

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

Daniel Müllner. 2013. fastcluster: Fast hierarchical, agglomerative clustering routines for r and python. *Journal of Statistical Software*, 53(9):1–18.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference*

on Computational Linguistics, pages 2340–2353, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Alicia Parrish, William Huang, Omar Agha, Soo-Hwan Lee, Nikita Nangia, Alexia Warstadt, Karmanya Aggarwal, Emily Allaway, Tal Linzen, and Samuel R. Bowman. 2021. Does putting a linguist in the loop improve NLU data collection? In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4886–4901, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mohammad Pezeshki, Oumar Kaba, Yoshua Bengio, Aaron C Courville, Doina Precup, and Guillaume Lajoie. 2021. Gradient starvation: A learning proclivity in neural networks. *Advances in Neural Information Processing Systems*, 34:1256–1272.

Pouya Pezeshkpour, Sarthak Jain, Sameer Singh, and Byron Wallace. 2022. Combining feature and instance attribution to detect artifacts. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1934–1946, Dublin, Ireland. Association for Computational Linguistics.

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

Shachar Rosenman, Alon Jacovi, and Yoav Goldberg. 2020. Exposing Shallow Heuristics of Relation Extraction Models with Challenge Data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3702–3710, Online. Association for Computational Linguistics.

Shiori Sagawa, Aditi Raghunathan, Pang Wei Koh, and Percy Liang. 2020. An investigation of why overparameterization exacerbates spurious correlations. In *International Conference on Machine Learning*, pages 8346–8356. PMLR.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

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

Shibani Santurkar, Dimitris Tsipras, and Aleksander Madry. 2021. BREEDS: Benchmarks for subpopulation shift. In *International Conference on Learning Representations*.

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

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

Roy Schwartz and Gabriel Stanovsky. 2022. On the limitations of dataset balancing: The lost battle against spurious correlations. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2182–2194, Seattle, United States. Association for Computational Linguistics.

Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. 2020. The pitfalls of simplicity bias in neural networks. *Advances in Neural Information Processing Systems*, 33:9573–9585.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Anders Søgaard, Sebastian Ebert, Jasmijn Bastings, and Katja Filippova. 2021. We need to talk about random splits. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1823–1832, Online. Association for Computational Linguistics.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293, Online. Association for Computational Linguistics.

Alon Talmor, Ori Yoran, Ronan Le Bras, Chandra Bhagavatula, Yoav Goldberg, Yejin Choi, and Jonathan Berant. 2021. CommonsenseQA 2.0: Exposing the

limits of AI through gamification. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. 2020. Measuring robustness to natural distribution shifts in image classification. *Advances in Neural Information Processing Systems*, 33:18583–18599.

Lifu Tu, Garima Lalwani, Spandana Gella, and He He. 2020. An empirical study on robustness to spurious correlations using pre-trained language models. *Transactions of the Association for Computational Linguistics*, 8:621–633.

Prasetya Utama, Nafise Sadat Moosavi, Victor Sanh, and Iryna Gurevych. 2021. Avoiding inference heuristics in few-shot prompt-based finetuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9063–9074, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Joe H Ward Jr. 1963. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244.

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

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

Yadollah Yaghoobzadeh, Soroush Mehri, Remi Tachet des Combes, T. J. Hazen, and Alessandro Sordoni. 2021. Increasing robustness to spurious correlations using forgettable examples. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3319–3332, Online. Association for Computational Linguistics.

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

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

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1298–1308, Minneapolis, Minnesota. Association for Computational Linguistics.

A Experimental Details

A.1 Datasets

We experiment with four large datasets: *QQP*, *MultiNLI*, WANLI and *ANLI*. We also run a hyperparameter search on *SST-2*, and evaluate model performance on *HANS* and *PAWS*. Sizes of the different datasets are reported in Tab. 6. Our implementation loads all datasets from Huggingface Datasets Hub using the *datasets* python library (Lhoest et al., 2021). All datasets are for English tasks.

QQP We experiment with the Quora Question Pairs[13] (*QQP*) dataset using the version released under the GLUE benchmark (Wang et al., 2018). *QQP* is a dataset for the task of predicting whether pairs of questions have the same intent, i.e., if they are duplicates or not. The dataset is based on actual data from Quora.

Natural Language Inference (NLI) The task of natural language inference involves predicting the relationship between a premise and hypothesis sentence pair. The label determines whether the hypothesis entails, contradicts or is neutral to the premise.

MultiNLI We experiment with the multi-genre MultiNLI dataset (Williams et al., 2018), which was crowdsourced by tasking annotators to write hypotheses to a given premise for each of the three labels. MultiNLI contains ten distinct premise genres of written and spoken data: (Face-to-face, Telephone, 9/11, Travel, Letters, Oxford University Press, Slate, Verbatim, Government and Fiction, of which five are included in the train and dev-matched sets. We don't use the dev-mismatched set in our experiments. We use the version released under the GLUE benchmark (Wang et al., 2018).

Adversarial NLI We experiment with Adversarial NLI (ANLI) (Nie et al., 2020), a large-scale human-and-model-in-the-loop natural language inference dataset collected over multiple rounds, using BERT (Devlin et al., 2019) and ROBERTA (Liu et al., 2019b) as adversary models. Although each of the dataset's rounds can be used as separate evaluation settings (e.g., training on the first round and testing on the second), the data collected over all rounds can also be concatenated and used for training and evaluation; both settings were used in the original paper. In our experiments we take the concatenation approach.

WANLI We experiment with WANLI (Liu et al., 2022b), an NLI dataset collected based on worker and AI collaboration. WANLI was created by identifying examples with challenging reasoning patterns in *MultiNLI* and using a LLM to compose new examples with similar patterns. The generated examples were then automatically filtered, and finally revised and labeled by human crowdworkers. WANLI is more challenging to models than *MultiNLI*, and using WANLI instances for training was shown to improve out-of-distribution generalization.

SST-2 We run a hyperparameter search on *SST-2*. The Stanford Sentiment Treebank (Socher et al., 2013) is a sentiment analysis corpus with fully labeled parse trees for single sentences extracted from movie reviews. *SST-2* refers to a binary classification task on sentences extracted from these parse tress (negative or somewhat negative vs somewhat positive or positive, with neutral sentences discarded). We use the version of *SST-2* released under the GLUE benchmark (Wang et al., 2018).

HANS We evaluate models on *HANS* (Heuristic Analysis for NLI Systems; McCoy et al. 2019), a challenge set used to assess whether NLI models adopt invalid syntactic heuristics that succeed for the majority of NLI training examples (e.g., lexical overlap implies that the label is entailment), instead of learning more generalizable solutions. *HANS* contains many *entailment* examples that support these heuristics, and many *non-entailment* examples where such heuristics fail. When evaluating NLI models that were trained with 3-way labels (as in *MultiNLI*), we map *contradiction* or *neutral* predictions to the *non-entailment* label. *HANS* was created by automatically filling in words in templates devised by human experts.

PAWS We evaluate models on *PAWS* (Paraphrase Adversaries from Word Scrambling; Zhang et al. 2019), a challenge set for the paraphrase identification task that focuses on non-paraphrase pairs with high lexical overlap. Challenging pairs are generated by controlled word swapping and back translation, followed by fluency and paraphrase judgments by human raters. We evaluate models on the test set of the $PAWS_{\text{Wiki}}$ dataset.

[13] https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs

		MultiNLI				*QQP*				WANLI				*ANLI*			
		Orig.	Cart.	ParIn	Mino.	Orig.	Cart.	ParIn	Mino.	Orig.	Cart.	ParIn	Mino.	Orig.	Cart.	ParIn	Mino.
Train	*full*	$91.1_{0.1}$	$64.4_{0.6}$	$81.4_{0.3}$	$74.3_{1.0}$	$93.0_{0.0}$	$66.0_{0.2}$	$81.3_{0.1}$	$77.6_{0.7}$	$77.1_{0.5}$	$26.4_{2.9}$	$62.6_{2.0}$	$61.7_{1.3}$	$67.5_{0.5}$	$35.0_{1.2}$	$50.0_{0.6}$	$58.3_{0.6}$
	rand	$91.1_{0.1}$	$64.7_{0.7}$	$81.5_{0.4}$	$75.1_{0.6}$	$92.8_{0.1}$	$65.5_{0.3}$	$81.1_{0.4}$	$77.4_{0.1}$	$77.3_{0.4}$	$26.4_{2.1}$	$60.8_{4.0}$	$61.0_{2.6}$	$67.6_{0.4}$	$34.6_{1.0}$	$49.1_{1.2}$	$58.5_{0.8}$
	bias	$89.4^{*}_{0.6}$	$56.6_{0.3}$	$71.8_{0.5}$	$57.5_{0.5}$	$89.4^{*}_{1.8}$	$52.6_{0.4}$	$63.9_{0.3}$	$36.8_{0.9}$	$76.6^{*}_{0.9}$	$22.2_{1.1}$	$49.6_{1.0}$	$31.8_{1.5}$	$60.2^{*}_{2.6}$	$12.9_{0.6}$	$28.3_{1.3}$	$21.4_{0.8}$

Table 5: Accuracy of our approach with DEBERTA-LARGE models. Different rows correspond to different training schemes: the full dataset (*full*), a biased subset (*bias*) and a random subset the size of *bias* (*rand*). Column groups correspond to different datasets. Individual columns represent testing schemes: the original validation/test set (Orig.) and the *anti-biased* test splits: *dataset cartography* (Cart.), *partial-input* (ParIn) and *minority examples* (Mino.). Reported values are averaged across three random seeds, with standard deviation as subscripts. Results in the last row (*bias*) are of training on the *biased* split and testing on the respective *anti-biased* split, except for Orig. values (marked with $*$), which are averaged over runs on all three *biased* splits.

	Train	Validation	Test
QQP	363,846	40,430	-
MultiNLI	392,702	9,815	-
ANLI	162,865	3,200	3,200
WANLI	102,885	5,000	-
SST-2	67,349	872	-
HANS	-	-	30,000
PAWS	-	-	8,000

Table 6: Datasets sizes. Development set in MultiNLI is the matched validation set (we did not use the mismatched validation set).

A.2 Experimental Settings

We experiment with the BASE and LARGE variants of ROBERTA (Liu et al., 2019b) and DEBERTA (He et al., 2021). Our implementation and pretrained model checkpoints use the Huggingface Transformers library (Wolf et al., 2020). For DEBERTA, we use the latest v3 checkpoints. When partitioning datasets to *biased* and *anti-biased* subparts, we use the training dynamics and representations of a ROBERTA-BASE model. We create *biased* training and *anti-biased* test sets based on a single run of the model. All further experiments (e.g., training DEBERTA-LARGE on *biased* instances and testing it on *anti-biased* instances) are run with 3 random seeds, using the same train and test splits.

Bias-amplified split sizes Tab. 7 reports the sizes of the bias-amplified *biased* train and *anti-biased* test splits created based on each of the three methods (Sec. 3) we experimented with.

Hyperparameters For fine-tuning, we did not optimize the hyperparameters and instead used parameters that were included in the hyperparameter search on down-stream tasks from the original papers, except for training LARGE models for 5 epochs instead of 10. We also used an early-stopping patience threshold of 3 epochs. We report all fine-tuning hyperparameters in Tab. 8 and Tab. 9.

Average runtimes For ROBERTA-BASE, each train run was performed on a single RTX 2080Ti GPU (10GB). For all other models, each train run was performed on a single Quadro RTX 6000 GPU (24GB). We report average runtimes (training and inference combined) in Tab. 10.

B Additional Results

B.1 Main Results for DEBERTA

Tab. 5 shows our results for DEBERTA-LARGE for the experiment described in Sec. 4.1.

C Clustering Algorithm for Detecting Minority Examples

Minority examples (Tu et al., 2020; Sagawa et al., 2020) are often detected by searching for spurious features correlated with one label in the instances of another label (e.g., high word overlap between two non-paraphrase texts). Motivated by recent work that leverages [CLS] token similarity in fine-tuned models between different instances (Liu et al., 2022b; Pezeshkpour et al., 2022), we proposed a model-based clustering approach to automatically detect minority examples (Sec. 3)

Our approach is based on simple analyses applied to the clustering of a given dataset's [CLS] model representations. In this work we used the deep clustering algorithm described in Sec. 3, DEEPCLUSTER (Caron et al., 2018), to perform the clustering. In this appendix we provide more details on the algorithm (App. C.1), its implemen-

	MultiNLI		*QQP*		WANLI		*ANLI*	
	train	test	train	test	train	test	train	test
dataset cartography	309,873	2,070	297,735	7,346	89,402	656	134,068	566
partial-input	309,873	2,070	297,735	7,346	89,402	656	134,068	566
minority examples	309,873	2,044	297,735	7,462	89,402	637	134,068	938

Table 7: Sizes of the train and test bias-amplified splits created with each of the considered methods (Sec. 3). Since the number of *biased* train instances is induced by the clustering in the *minority examples* approach, but is a hyperparameter q for the two other approaches, we simply adjust q to create equally sized training sets for all three methods. We use the same q used for choosing *biased* **train** instances when choosing *anti-biased* **test** instances. We note that for the *minority examples* method, the training set clustering and the predicted test set clustering (based on a simple nearest neighbor classifier fitted on the training set) are two *different* clusterings, which can result in different proportions of *minority examples* between the train and test sets. This explains the difference in the amounts of *anti-biased* test instances between *minority examples* and the other two methods.

Hyper-parameter	BASE	LARGE
Warmup Ratio	0.06	0.06
Learning Rate	1e-5	1e-5
Learning Rate Decay	Linear	Linear
Batch Size	32	32
Max. Train Epochs	10	5
Early Stopping Patience	3	3

Table 8: Hyperparamets for finetuning ROBERTA.

Hyper-parameter	BASE	LARGE
Warmup Steps	100	100
Learning Rate	1.5e-5	1e-5
Learning Rate Decay	Linear	Linear
Batch Size	32	32
Max. Train Epochs	10	5
Early Stopping Patience	3	3

Table 9: Hyperparamets for finetuning DEBERTA.

Datasets	ROBERTA		DEBERTA	
	BASE	LARGE	BASE	LARGE
MultiNLI	8	8	-	10
QQP	8	8	-	10
WANLI	2	4	2	4
ANLI	4	5	-	5
SST-2	1	-	-	-

Table 10: Average runtimes for fine-tuning, in hours.

tation details (App. C.2), and the hyperparameter search we ran to select a good configuration (App. C.3). We also show preliminary results for using alternative clustering methods for detecting *minority examples* (C.4.1) and for the difficulty of the bias-amplified splits based on *minority examples* detected over different random seeds (App. C.4.2).

C.1 DEEPCLUSTER

DEEPCLUSTER alternates between grouping the model's representations with a standard clustering algorithm to produce pseudo-labels, and updating the parameters of the model by predicting these pseudo-labels. To apply DEEPCLUSTER to BERT-like models,[14] we consider a model fine-tuned on the dataset to be clustered.[15] We extract and cluster its [CLS] token representations using a standard clustering algorithm, and then perform one DEEPCLUSTER iteration by fine-tuning a new pretrained model with the pseudo-labels (instead of the dataset's gold labels) for one epoch.[16] We then cluster the representations from this second model to obtain the final clustering.

C.2 Implementation Details

As the standard clustering algorithm at the base of DEEPCLUSTER, we use Ward's method (Ward Jr, 1963), a popular hierarchical clustering algorithm which is deterministic and therefore stable across different runs, a quality which we found preferable. We use the fastcluster (Müllner, 2013) python implementation with the default settings.

[14] DEEPCLUSTER is used in the original paper for pretraining Computer Vision models.

[15] Importantly, the model is fine-tuned on the *task label*, rather than the clustering label (such labels do not exist a priori, but are generated automatically).

[16] We emphasize that this pretrained was never fine-tuned on any task labels, but only on the pseudo-labels.

Applying Ward's clustering to large-scale datasets We did not have resources with enough memory to cluster the entire training sets of *MultiNLI* and *QQP*, which contain more than 320k examples. We therefore approximate the clustering assignment by clustering a random sample of 50% of the training set, and then using a simple nearest-neighbor classifier to predict the assignments for the other 50%.[17]

Runtime Running DEEPCLUSTER requires (1) fine-tuning a model for 1 epoch and then extracting its representations, which takes 15–70 minutes on a GPU, and (2) clustering the representations on a CPU, which takes 40 minutes for WANLI and *ANLI*, and 3 hours for *MultiNLI* and *QQP*.

C.3 DEEPCLUSTER Hyperparameters

DEEPCLUSTER has three hyperparameters: the number of final clusters k, the number of pseudo-labels m for representation learning, and the Transformer layer from which [CLS] representations are extracted for clustering. We used $k = 10$ clusters for all datasets, and searched for a good configuration for the other two hyperparameters on *SST-2* (Socher et al., 2013), which were then used for experiments on all other datasets in the paper. We searched over $m \in \{10, 30, 50, 100, 300, 500, 1000, 1500, 3000\}$[18] and representations from the last four layers of ROBERTA-BASE.

For each set of hyperparameters, we applied the *minority examples* method to create *biased* training and *anti-biased* test splits, and trained two ROBERTA-BASE models—one on the *biased* train split, and a baseline model on an equally-sized random train subset, finally choosing the hyperparameters that lead to the largest performance drop on *anti-biased* test instances between the two. The best hyperparameters were the layer before last of ROBERTA-BASE (layer 11) and $m = 1500$.

C.4 Preliminary Results

C.4.1 Using Standard Clustering to Detect Minority Examples

Our preliminary experiments show that standard clustering algorithms applied to the [CLS] representations of models fine-tuned on the original task tend to create label-homogeneous clusters, i.e., they are less likely to cluster together instances from different labels. In Fig. 5 we show the average proportions of majority and minority instances within clusters for different clusterings of *SST-2* (which has two task labels) based on ROBERTA-BASE representations. We compare DEEPCLUSTER and two standard clustering algorithms: K-Means and Ward's method. We find that the clusters of standard methods contain, on average, less than 5% minority label instances, while clusters based on DEEPCLUSTER are more label-diverse and contain 15% minority label instances. When inspecting how many individual clusters contain more than 10% minority label instances, we find that for both standard methods **only one cluster** (out of 10) meets this threshold, whereas there are **6 such clusters** with DEEPCLUSTER.

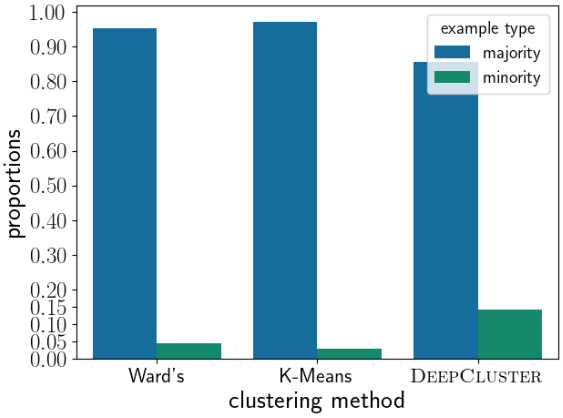

Figure 5: The mean proportions of majority and minority label instances within clusters for different clusterings of *SST-2*, based on the [CLS] representations of ROBERTA-BASE fine-tuned on the dataset. [CLS] tokens are taken from the layer before last of the model.

C.4.2 Difficulty of Minority Examples in Bias-amplification Over Random Seeds

We ran a preliminary experiment on *SST-2* to examine whether the difficulty of the bias-amplified splits based on the *minority examples* method varies with the seed used to collect data representations. We clustered *SST-2* using DEEPCLUSTER based on representations of ROBERTA-BASE. We used 3 different seeds to fine-tune the model and run DEEPCLUSTER, and created a bias-amplified split from each resulting clustering. We then examined the performance drops between a ROBERTA-BASE model trained on the *biased* vs. random split (as in the hyperparameter search; see App. C.3).

[17] We fit the classifier on the clustered sample's representations as inputs and clustering assignments as output labels.

[18] The number of pseudo-labels m can differ from the desired number of final clusters k. Setting $m \gg k$ yielded better results in the original paper.

The mean absolute performance drop was -16.7, with a standard deviation of 5.9. This indicates that while there is variation between seeds, all clusterings produced challenging settings. We conclude that when seeking to create the *most* challenging splits, running a hyperparameter search over multiple seeds on the dataset the splits are created for would likely lead to better results. In this work, we did not optimize the clustering hyperparameters for each dataset, and therefore used one seed for all clusterings.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
section 2.3 and the limitations section.

☑ A2. Did you discuss any potential risks of your work?
the limitations section (paragraph 2).

☑ A3. Do the abstract and introduction summarize the paper's main claims?
2,3,4,5,6

☑ A4. Have you used AI writing assistants when working on this paper?
ChatGPT was occasionally used for assistance purely with the language of the paper (mostly to ask 'what are better ways to say "____"'), throughout all sections.

B ☑ Did you use or create scientific artifacts?

4.1, 4.2, A.1

☑ B1. Did you cite the creators of artifacts you used?
4.1, 4.2, A.1, A.2, C.2

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
We will discuss the license of the artifacts we created (as well as the artifacts that were used in their creation) with the release of our code and biased splits, with the publication of the paper.

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
For the artifacts we created (simple subsets of existing datasets) - section 2.3 and the limitations section. Existing artifacts (datasets, models) were used in standard ways (evaluation, fine-tuning).

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
We used widely-cited datasets in ways that don't concern privacy/toxicity, and did not collect any new data.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
A.1

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
A.1, A.2

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

C ☑ Did you run computational experiments?

4, 5, 6

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
A.2, C.2

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
4.1, A.2, C.3

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
4, 5, 6

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
A.2, C.2

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.