Stubborn Lexical Bias in Data and Models

Sofia Serrano[1] **Jesse Dodge**[2] **Noah A. Smith**[12]
[1]Paul G. Allen School of Computer Science & Engineering, University of Washington
[2]Allen Institute for Artificial Intelligence
sofias6@cs.washington.edu, jessed@allenai.org, nasmith@cs.washington.edu

Abstract

In NLP, recent work has seen increased focus on spurious correlations between various features and labels in training data, and how these influence model behavior. However, the presence and effect of such correlations are typically examined feature by feature. We investigate the cumulative impact on a model of many such intersecting features. Using a new statistical method, we examine whether such spurious patterns in data appear in models trained on the data. We select two tasks—natural language inference and duplicate-question detection—for which any unigram feature on its own should ideally be uninformative, which gives us a large pool of automatically extracted features with which to experiment. The large size of this pool allows us to investigate the intersection of features spuriously associated with (potentially different) labels. We then apply an optimization approach to *reweight* the training data, reducing thousands of spurious correlations, and examine how doing so affects models trained on the reweighted data. Surprisingly, though this method can successfully reduce lexical biases in the training data, we still find strong evidence of corresponding bias in the trained models, including worsened bias for slightly more complex features (bigrams). We close with discussion about the implications of our results on what it means to "debias" training data, and how issues of data quality can affect model bias.

1 Introduction

Machine learning research today, including within NLP, is dominated by large datasets and expressive models that are able to take advantage of them. At the same time, as the scale of training data has grown, this explosion of data has come at the expense of data *curation*; for many of the datasets currently in use today, human oversight of the full breadth of their contents has become unrealistic. This makes it more likely that training datasets contain undesirable associations or shortcuts to learning intended tasks. Many cases are attested (e.g., Tsuchiya, 2018; Gururangan et al., 2018; Poliak et al., 2018; McCoy et al., 2019; Rudinger et al., 2018; Stanovsky et al., 2019; Davidson et al., 2019; Sap et al., 2019), and we suspect a vast number of these so-called "spurious correlations" remain undetected.

One question is whether these unintended biases in the training data propagate to models trained on that data. Recent work has found mixed results on this point (Steed et al., 2022; Joshi and He, 2022). We begin by introducing an approach to testing for undesirable model biases that can operate using existing held-out data, even though that data might itself have spurious correlations. In particular, we repurpose the classic permutation test to examine whether observed differences in model performance between instances exhibiting more common feature-label pairings and those exhibiting less common feature-label pairings are statistically significant.

For our experiments, we focus on the simplest kind of feature-label association: correlations between lexical features and task labels. We select two tasks (natural language inference and duplicate-question detection) for which any such lexical feature should be uninformative on its own. Finding strong evidence that models finetuned on three different datasets have at least some of the same lexical biases that exist in their training data, we then examine the extent to which those biases are mitigated by lessening biases in the training data. To do this, we apply an optimization-based approach to reweighting the training instances. The approach brings uneven label distributions closer to uniform for thousands of different intersecting lexical features, many more than we use for our model bias evaluation, and still manages to have a strong effect on the most initially biased features despite our reweighting approach not focusing on those

Findings of the Association for Computational Linguistics: ACL 2023, pages 8131–8146
July 9-14, 2023 ©2023 Association for Computational Linguistics

in particular. We then finetune new models on those (reweighted) datasets. We find that although model bias lessens somewhat when we do this, we still find strong evidence of bias. Surprisingly, this holds even when we consider models that make use of no pretraining data.

We close with a discussion of possible factors contributing to these results. We first note that perhaps the continued relative lack of variety of minority-class examples containing certain features hinders the reweighted models' ability to generalize their recognition of those less-common feature-class pairs, even though the combined weight given to those few instances in the loss function is increased. However, when we examine the effect of our reweighting on higher-order features (namely, bigrams), we see another problem: the same reweighting that mitigates associations between unigrams and any particular label actually strengthens associations between bigrams and certain labels in data. Based on this observation, we arrive at two conclusions: (1) simultaneously reducing bias across features of different levels of granularity for natural-language data is likely not feasible, and (2) even if we aim to mitigate model bias *only* with respect to simple features, if we do so by reweighting the data, the high-capacity models used in modern NLP are still capable of learning the spurious correlations of the original unweighted data through associations that remain encoded in more complex features even after reweighting. We conclude that bias reduction in NLP cannot be cast purely as a "data problem," and solutions may need to focus elsewhere (e.g., on models).

2 What Do We Mean by Bias?

The term "bias" is polysemous, having been adopted by different communities to mean different things, from historically rooted social inequity to skewed model evaluations (Mehrabi et al., 2021) to techniques that help with supervised class imbalance in labels (Chen et al., 2018). In our work, we use "bias" to mean correlations between individual input features and task labels. This framework is fairly general, but our focus in this work is natural language data. Therefore, as an example to illustrate our definition of bias, we will refer to correlations between the presence of individual word types in the input (unigrams) and a given label in a classification task.

More formally, consider a task of mapping inputs in $\mathcal{X}$ to labels in $\mathcal{Y}$. We assume a training dataset $\mathcal{D} = \langle (x_i, y_i) \rangle_{i=1}^{n}$, each $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$. We are particularly interested in a designated collection of d binary features on $\mathcal{X}$, the jth of which is denoted $f_j : \mathcal{X} \rightarrow \{0, 1\}$. For example, f_j might be the presence of the word "nobody" in an instance. Let $f_{j,i}$ be shorthand for $f_j(x_i)$ (e.g., whether instance x_i contains the word "nobody" ($f_j(x_i) = 1$) or not ($f_j(x_i) = 0$)).

Introducing random variable notation, we can characterize $\mathcal{D}$ by its empirical conditional distribution over labels given each feature, such that for all $y \in \mathcal{Y}$,

$$\hat{p}(Y = y \mid F_j = 1) = \frac{\sum_i \mathbf{1}\{f_{j,i} = 1 \wedge y_i = y\}}{\sum_i \mathbf{1}\{f_{j,i} = 1\}}.$$

If the conditional distribution of output labels given the presence of a particular lexical feature is very different from the overall label distribution in the data, we consider that feature to be biased in the training data.

3 Measuring Bias in Model Performance and Data

Recall that when $\hat{p}(Y = y \mid F_j = 1)$ is close to 1, it means feature j is correlated with label y in a given dataset. Let us denote the set of examples that contain feature j and have the label most strongly associated with feature j in $\mathcal{D}$ by $\mathcal{U}_j$, which we call the "usual-labels" set. Then, denote the examples that contain j but have a *different* label by $\mathcal{N}_j$, which we call the "unusual-labels" set.

To build intuition, the accuracy of the model on instances which contain feature j is the accuracy over the union $\mathcal{U}_j \cup \mathcal{N}_j$. However, to measure if the model is picking up bias from the data, we will measure accuracy over $\mathcal{U}_j$ and $\mathcal{N}_j$ separately. To maximize accuracy on $\mathcal{U}_j \cup \mathcal{N}_j$ the model would be justified in disproportionately labeling instances containing f_j with y, so we can't use accuracy by itself to measure model bias. Instead, the key idea here will be to look for differences in error rates between instances whose labels align with features' training biases (the "usual-labels" set), and instances whose labels do not.

If the model has learned a biased representation of the data, we expect it to have higher accuracy on the "usual-labels" set, $\mathcal{U}_j$. On the other hand, if the model hasn't learned that bias, we would expect the correct predictions to be uniformly distributed between $\mathcal{U}_j$ and $\mathcal{N}_j$. We use this as the basis for a hypothesis test: the null hypothesis H_0 is that the accuracy of model is the same on both sets $\text{ACC}(\mathcal{U}_j) = \text{ACC}(\mathcal{N}_j)$, and the alternative hypothesis H_1 is that $\text{ACC}(\mathcal{U}_j) > \text{ACC}(\mathcal{N}_j)$. That is, if the errors are distributed uniformly at random, how likely is it that $\mathcal{U}_j$ would have *at least* its observed number of correct instances?

3.1 Permutation Test

Given a model's accuracy on $\mathcal{U}_j$ and $\mathcal{N}_j$, and the size of the two sets, we can calculate the p-value for this hypothesis test exactly using the permutation test (Phipson and Smyth, 2010). Our null hypothesis is that the errors are uniformly distributed between $\mathcal{U}_j$ and $\mathcal{N}_j$, so the permutation test calls for randomly shuffling whether a given instance is correctly labeled, while not changing the number of instances in each category *or* the model's overall accuracy on the set union, both of which change the shape of the distribution of correct instances that we'd expect to see, but neither of which is the property for which we're testing. As there are finitely many ways to shuffle whether a given instance is correctly labeled, this test also has the benefit of having a closed form, giving us an exact p-value.[1]

3.2 Calculating Bias over Multiple Features

In the previous section we described how we could use a permutation test for a single feature f_j. Here we describe how to apply this to the full dataset. We define $\mathcal{U}$ as $\cup_j \mathcal{U}_j$ and $\mathcal{N}$ as $\cup_j \mathcal{N}_j$ for 50 features f_j per distinct label (namely, those that demonstrate the highest association with that label in the training data), so 100 or roughly 150 features f_j total depending on whether the dataset is 2- or 3-class ("roughly" because some features are among the most associated for two classes in 3-way classification). Given that each example x_i includes multiple features (e.g., $f_{j,i} = 1 \wedge f_{k,i} = 1$) it's possible for example x_i to have label y, which is the "usual-labels" for f_j but an "unusual-labels" for f_k. When this happens, we add it to both sets $\mathcal{U}$ and $\mathcal{N}$, meaning that their intersection is not necessarily empty. Pooling examples in this way allows us to run a single hypothesis test for whether or not the model learns bias from the dataset, avoiding the multiple-comparisons issue of running one hypothesis test for each feature. This procedure is described in Figure 1.

4 Applying the Test

Here we shift our focus to particular tasks and datasets, in order to apply our test in practice.

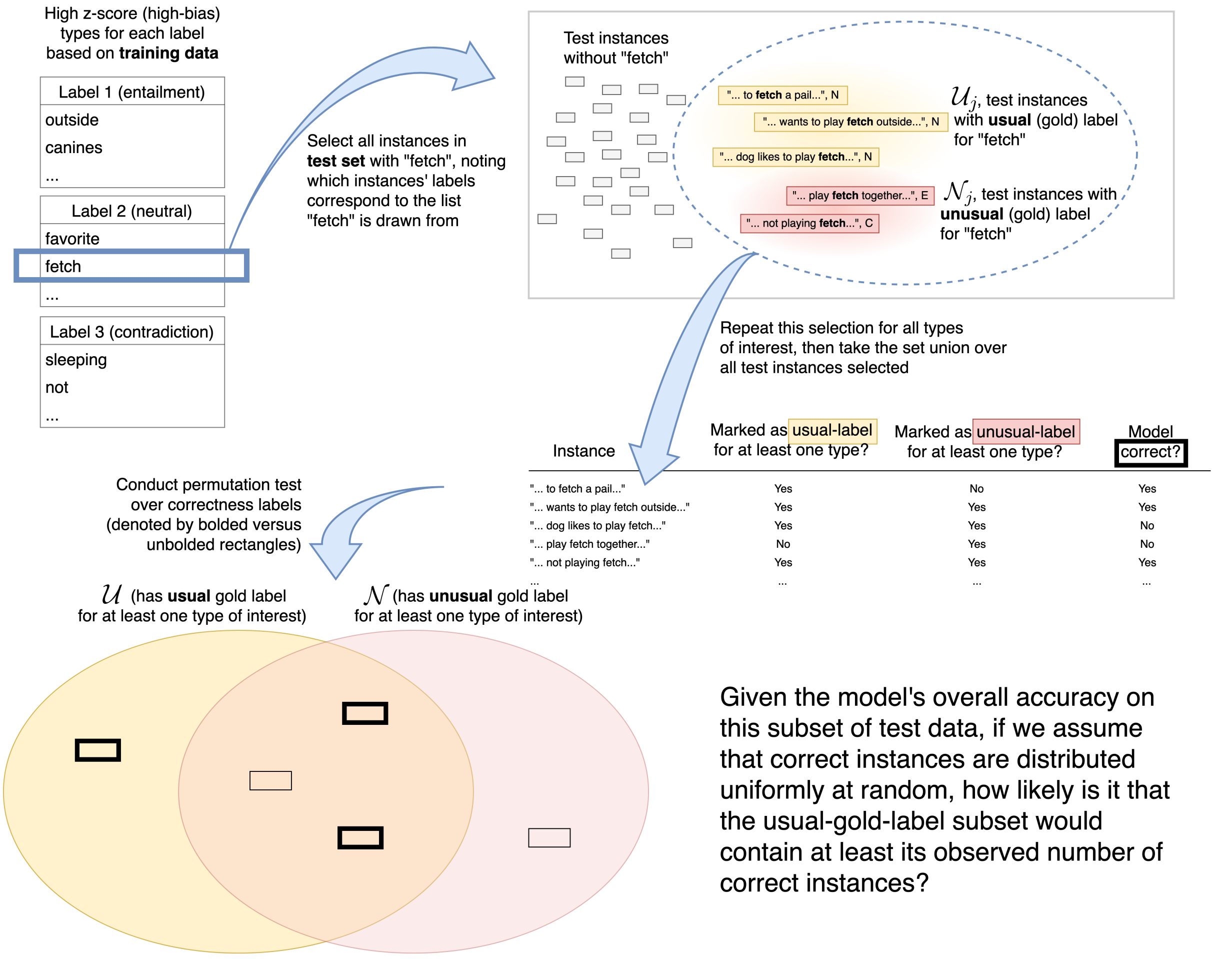
4.1 Determining Biased Features (and Tasks)

For our experiments, we want a large volume of features that should ideally exhibit no correlation with labels. In order to get a large number of features, we'd like them to be simple and easy to automatically detect, so unigram features again come to mind, guiding our selection of tasks and datasets for experiments.

When is the association of unigram features with a particular label a problem? While previous work has argued that the presence of an individual word type in a given instance, by itself, does not provide enough information to predict the label for *any* ideal task that requires an understanding of natural language (Gardner et al., 2021), in this work we consider this argument only as it relates to two tasks where such a position is relatively uncontroversial: natural language inference, and duplicate-question detection.

Consider the task of natural language inference (NLI), where the input consists of two sentences (premise and hypothesis), and the correct label is a human annotation indicating whether the premise entails the hypothesis, contradicts it, or neither. Continuing our example from section 2, if $f_{j,i} = 1$, then the word "nobody" appears somewhere in example x_i (premise, hypothesis, or both). Given these definitions of the task and the features, $f_{j,i} = 1$ by itself is uninformative for predicting y_i (intuitively, we don't learn any information about whether or not the premise entails the hypothesis by knowing that the word "nobody" appears somewhere in the input). However, it has been shown that in the SNLI dataset (Bowman et al., 2015) $f_j = 1$ almost perfectly predicts the label, in both the training and test sets (for example, in the training set, 2368 instances with $f_j = 1$ have a label of "contradiction" and only 13 don't). Thus, this is an example of a "spurious correlation" (or, bias in the data).

[1] For simplicity, we assume here that the model has an equal likelihood of guessing any of the output classes. In practice, this is approximately accurate for the data on which we experiment, though this assumption could be removed in principle by multiplying each permutation by a corresponding probability.


High z-score (high-bias) types for each label based on **training data**
Label 1 (entailment)
outside
canines
...
Label 2 (neutral)
favorite
fetch
...
Label 3 (contradiction)
sleeping
not
...
Test instances without "fetch"
"... to **fetch** a pail...", N
"... wants to play **fetch** outside...", N
"... dog likes to play **fetch**...", N
"... play **fetch** together...", E
"... not playing **fetch**...", C
$\mathcal{U}_j$, test instances with **usual** (gold) label for "fetch"
$\mathcal{N}_j$, test instances with **unusual** (gold) label for "fetch"
$\mathcal{U}$ (has **usual** gold label for at least one type of interest)
$\mathcal{N}$ (has **unusual** gold label for at least one type of interest)

Select all instances in **test set** with "fetch", noting which instances' labels correspond to the list "fetch" is drawn from

Repeat this selection for all types of interest, then take the set union over all test instances selected

Instance	Marked as usual-label for at least one type?	Marked as unusual-label for at least one type?	Model correct?
"... to fetch a pail..."	Yes	No	Yes
"... wants to play fetch outside..."	Yes	Yes	Yes
"... dog likes to play fetch..."	Yes	Yes	No
"... play fetch together..."	No	Yes	No
"... not playing fetch..."	Yes	Yes	Yes
...	...	...	...

Conduct permutation test over correctness labels (denoted by bolded versus unbolded rectangles)

Given the model's overall accuracy on this subset of test data, if we assume that correct instances are distributed uniformly at random, how likely is it that the usual-gold-label subset would contain at least its observed number of correct instances?

Figure 1: The setup of the permutation test that we use to test for bias in models trained on biased data, which in this figure uses word types as features and natural language inference as the underlying task.

4.2 Applying the Test to Models

We now apply the described permutation test to finetuned models. For each of SNLI (Bowman et al., 2015), QNLI (Wang et al., 2018), and QQP,[2] we finetune three pretrained RoBERTa-large models (Liu et al., 2019) with different random seeds on their training sets. We use a learning rate of 2×10^{-6} and finetune for 15 epochs using a single GPU with 12GB memory.

Following the argument by Gardner et al. (2021) that unigram features for these kinds of theoretically complex tasks should ideally be uninformative in isolation, we use lexical types as our bias evaluation features. For the purpose of this calculation, each label will contribute the 50 features that have the strongest correlation with it (as calculated by z-score, again following Gardner et al., 2021) in the lowercased training data, excluding stop words, since they tend to receive high z-scores due to appearing in such an overwhelming number of instances.[3] We then select all test instances with one or more of those types present as our evaluation set for our permutation test. For models finetuned on SNLI and QQP, we find p-values of at most 2.3×10^{-17} (see "Trained on uniform" rows of Table 2), indicating very strong evidence that—as expected—these models reflect the bias associated with types with high z-scores in the training set. For QNLI, we see mixed results depending on our random seed, with p-values of 0.0057, 0.024, and 0.053 for our three finetuned models. (Worth noting is the fact that, as we will see later in Section 5.1, QNLI has the lowest overall feature-label bias of any of these three datasets.) Still, we see enough of these models demonstrating bias to merit investigating why this occurs.

[2] Quora Question Pairs dataset (QQP): `data.quora.com/First-Quora-Dataset-Release-Question-Pairs`

[3] In section A.1, for illustration purposes, we include the resulting list of 50 lexical types per label for SNLI.

5 Where Does that Bias Come From?

Having established that there is often similar bias in the finetuning data and models trained on that data, we consider that the finetuning data is not necessarily the source of the bias in the model. For example, the bias could come from the pretraining data as well. With that in mind, how might we check the impact of the finetuning data specifically?

5.1 Intervening on the Data by Balancing It

Our strategy is to intervene on the data to lessen lexical bias.[4] While modifying the data is only one family of approaches towards reducing eventual bias of a learned model (see, for example model-based strategies such as those proposed by Clark et al., 2019, or Karimi Mahabadi et al., 2020), recall that our goal here is to investigate the effect of the finetuning data on the rest of the training setup, so for our purposes we keep the rest of the training procedure the same.

Prior work has explored different ways of intervening on data, such as manual data augmentation (Zhao et al., 2018; Zhang and Sang, 2020; Gowda et al., 2021; Lee et al., 2021), or occluding bias in the original data (Feldman et al., 2015), but along very few different axes of bias. Other work augments minority-class data for the purpose of addressing class imbalance (Chawla et al., 2002). Yet others have taken the approach of generating new data to augment the existing data in ways that counteract certain biases (Wu et al., 2022). However, this last work relies on model-generated text, which, as Wu et al. (2022) themselves acknowledge, could differ from human-generated text in ways that aren't immediately obvious (Zellers et al., 2019).

In order to avoid potential new artifacts introduced by using machine-generated training data, and to improve the label balance in aggregate for a large volume of features simultaneously, we reweight existing training data such that in expectation, the disproportionate association of lexical features with certain labels is decreased. Reweighting data to remove bias is not a new idea—Kamiran and Calders (2012) do this through downsampling—but typically such approaches have considered at most a handful of different axes of bias. Some existing work, namely Byrd and Lipton (2018) and Zhai et al. (2023), has pointed out the limitations of approaches based on reweighting data, but again based on reweighting along comparatively few axes (in the case of the former) or on simpler model architectures than we consider here (in the case of the latter), so in the absence of a viable alternative meeting our requirements, we proceed with reweighting as our form of intervention for our experiments.

Typically, training datasets like $\mathcal{D}$ are treated as i.i.d., representative samples from a larger population. Formally, we instead propose to *weight* the instances in $\mathcal{D}$, assigning probability q_i to instance i, such that, $\forall j, \forall y \in \mathcal{Y}$,

$$\frac{\sum_i q_i \cdot \mathbf{1}\{f_{j,i} = 1 \land y_i = y\}}{\sum_i q_i \cdot \mathbf{1}\{f_{j,i} = 1\}} = \frac{1}{|\mathcal{Y}|} \tag{1}$$

From here on, we denote the lefthand side of Equation 1 as $q(y \mid F_j = 1)$. Note that, for simplicity, we assume a uniform distribution over labels as the target, though our methods can be straightforwardly adapted to alternative targets.

Given an algorithm that produces a weighting $q_1, \ldots, q_n$ for dataset $\mathcal{D}$, we quantify its absolute error with respect to Equation 1 as

$$\begin{aligned}\text{Err}(q) &= \frac{1}{(\text{number of features}) \cdot |\mathcal{Y}|} \cdot \\ &\quad \sum_j \sum_{y \in \mathcal{Y}} \left| q(y \mid F_j = 1) - \frac{1}{|\mathcal{Y}|} \right|\end{aligned}$$

How do we choose these q_i values? We can state the general problem as a constrained optimization problem.[5] We seek values $q_1, \ldots, q_n$ such that:

$$\sum_{i=1}^{n} q_i = 1 \tag{2}$$

$$q_i \geq 0, \ \forall i \tag{3}$$

$$q(y \mid F_j = 1) - \frac{1}{|\mathcal{Y}|} = 0, \ \forall j, \forall y \in \mathcal{Y} \tag{4}$$

(The constraints in the last line are derived from Equation 1; strictly speaking one label's constraints are redundant and could be removed given the sum-to-one constraints.)

Using this setup, we seek a vector q that satisfies the constraints. We do this by minimizing the

[4] Note, we do not describe our approach as "removing bias," as natural language data in general is biased to some extent; see the argument made by Schwartz and Stanovsky (2022).

[5] The slightly simplified formulation we present here for ease of reading only takes into account cases where feature j appears somewhere in our data, but Equation 4 can be straightforwardly modified by multiplying it by the denominator of $q(y \mid F_j = 1)$ to account for this.

sum of squares of the left side of Equation 4; the approach is simplified by a reparameterization:

$$q_i = \frac{\exp z_i}{\sum_i \exp z_i}$$

This is equivalent to optimizing with respect to unnormalized weights (z_i) that are passed through a "softmax" operator, eliminating the need for the constraints in Equations 2 and 3. Once we have q, we multiply each x_i's contribution to the loss during training by $q_i \cdot |\mathcal{D}|$.

We apply this algorithm to reweight the following training datasets: SNLI (Bowman et al., 2015), MNLI (Williams et al., 2018), QNLI (Wang et al., 2018), and QQP. In contrast to the <200 features per dataset that we use for evaluation of bias in models, when reweighting data, we used all types that appeared at least 100 times in their corresponding training data as features, and we denoted an "instance" as the concatenation of a paired premise and hypothesis (or, for QQP, the concatenation of the two questions). We removed features from consideration if they did not have at least one document in the dataset for each of their labels.[6]

We see in Table 1 that by solving for distributions q over the different datasets as described, we successfully reduce $\text{Err}(q)$ compared to the initial uniform weighting for all datasets except MNLI.[7] This leaves us with three successfully reweighted datasets with lessened unigram bias overall, and we can use these to investigate possible reduction of lexical bias compared to their original, uniformly-weighted counterparts. We confirm that for the high-z-score features used for model bias evaluation for each of these three, their label balance in the data either improves (often dramatically) or stays comparable as a result of our reweighting q. (Here and elsewhere, we use "label balance" of a feature to refer to the average absolute difference between its empirical label distribution in the training data and the overall label distribution of the training data, averaging elementwise over each possible label.) For example, see Figure 2 for the change that our reweighted q makes in improving the label distributions of our original high-z-score features from SNLI that we use for evaluation.

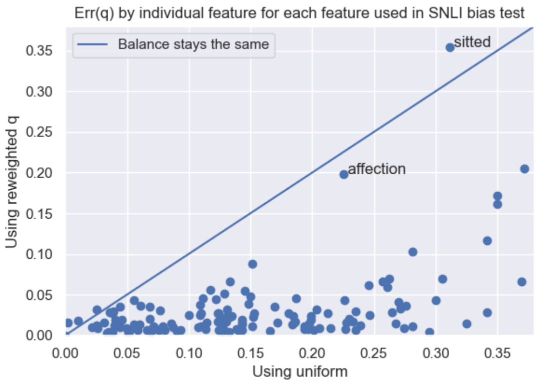


Figure 2: Label balance of the 137 lexical features used in our *model* bias evaluation for SNLI (since a handful of the highest z-score features in the training data didn't appear in the test set), using a uniform weighting and reweighed using q. q produces a lower $\text{Err}(q)$ for most of these features and is comparable for most of the remaining few, even considering that the reweighting was with respect to all 3,866 features. We have labeled the only two features that go against this pattern.

5.2 Impact when Finetuning on Reweighted Data

We now consider what happens when we finetune models on that data. We finetune RoBERTa-large models using new random seeds and all the same hyperparameters as before, only this time on training data reweighted using the new q distributions. We see similar validation accuracies (a point or so of difference), indicating that this reweighting has a small effect on overall performance, even though the validation sets may contain similar biases to their corresponding training sets and therefore benefit models that leverage those biases.

The results of rerunning our model bias evaluation are listed in the top half of Table 2. While we do see an increase in p-values, indicating weaker evidence of bias than for models trained on the uniformly-weighted training data, for both SNLI and QQP, we are still left with very strong evidence of bias (p-values of at most 1.2×10^{-5}). A natural question that we might ask is whether we can attribute this remaining bias to the pretraining data.

To test whether we see the same patterns in the absence of any other training data, we also train two bidirectional three-layer LSTMs per dataset from scratch (i.e., no pretraining and no pretraining data), one using uniform weighting and the other using q-reweighted.[8] As we can see in Table 2,

[6] This was not the case for any features in MNLI or QNLI, but applied to the word "recess" for SNLI, and the words "gobi" and "weakest" for QQP.

[7] MNLI is unusual among the datasets we studied in its remarkably low degree of lexical-feature bias to begin with, so it is perhaps not surprising that further lowering that bias across thousands of features proves difficult.

[8] To ensure no leaked signal from any other data, we initialized the word embeddings of the LSTMs to continuous

	$\lvert\mathcal{D}\rvert$	# Features	$\lvert\mathcal{Y}\rvert$	Err(Uniform) (↓)	Err(Adjusted q) (↓)
SNLI	549,367	3866	3	0.057	0.040
MNLI	392,376	6854	3	0.022	0.084
QNLI	104,743	3770	2	0.042	0.012
QQP	363,831	4386	2	0.154	0.047

Table 1: The average absolute difference between the empirical fraction of label y in instances with any particular unigram feature j and the total weight given to label y in the full training data, computed over all features and all their label values. Lower is better.

			p-value(s) for permutation test
Finetuned transformers	SNLI	Trained on uniform	$1.9 \times 10^{-35}, \{1.1, 2.2\} \times 10^{-23}$
		Trained on adjusted q	$\{1.2, 1.7, 3.2\} \times 10^{-14}$
	QNLI	Trained on uniform	$5.7 \times 10^{-3}, \{2.4, 5.3\} \times 10^{-2}$
		Trained on adjusted q	$\{3.7, 7.6, 2.6\} \times 10^{-1}$
	QQP	Trained on uniform	$2.4 \times 10^{-26}, 2.6 \times 10^{-20}, 2.3 \times 10^{-17}$
		Trained on adjusted q	$7.6 \times 10^{-20}, 5.9 \times 10^{-7}, 1.2 \times 10^{-5}$
From-scratch LSTM	SNLI	Trained on uniform	5.9×10^{-83}
		Trained on adjusted q	2.0×10^{-75}
	QNLI	Trained on uniform	3.1×10^{-61}
		Trained on adjusted q	1.6×10^{-10}
	QQP	Trained on uniform	Approx. 10^{-638}
		Trained on adjusted q	Approx. 10^{-762}

Table 2: Exact p-values for permutation tests conducted on different models, which check the probability that the usual-gold-label subset of the test data would have at least its observed accuracy if the instances guessed correctly by the model were distributed uniformly at random across the usual and unusual gold-label test subsets. The pretrained model used to initialize each finetuned transformer was RoBERTa-large, and for each pairing of a dataset and a uniform or adjusted weighting of its data in finetuning a transformer, we ran three separate random seeds to observe variance. For each dataset-weighting pairing in training LSTMs from scratch, we used a single random seed.

while there continues to be a rise in p-value with the switch to the reweighted q, the higher p-value is still vanishingly small. **All the models trained from scratch are biased.**

Of particular interest is the fact that the LSTMs trained on QNLI display strong evidence of bias, while the pretrained transformers that were finetuned on either version of QNLI (reweighted or not) were the only models that did not display strong evidence of bias. This indicates that at least in QNLI's case, bias has entirely separate causes than training data; for QNLI, it's only the models trained from scratch that display significant evidence of bias. This, along with the tiny p-values for the other LSTMs, indicates that there are still factors even in the reweighted data that contribute to bias.

	Err(Uniform)(↓)	Err(Adjusted q)(↓)
SNLI	0.059	0.122
QNLI	0.134	0.173
QQP	0.215	0.224

Table 3: The average absolute difference between the empirical distribution of label y (in the data) for instances with a **bigram** feature j and the overall distribution of label y given the full data (we perform this difference elementwise). The calculations over any row in this table are performed over 200 randomly selected bigrams j from that dataset, which are kept consistent across columns. Lower is better.

At first, this is surprising. Given that the LSTMs trained with the reweighted q distributions over data were exposed to no other data, why do they still exhibit bias? One possibility is issues of quality inherent to some unusual-label data. For example, consider the word "favorite" in SNLI, which has one of the highest z-scores for the "neutral" label. Even though nothing about the task of de-

bag-of-words embeddings (Mikolov et al., 2013) trained using their respective q-weighted training sets. We use a word embedding dimension of 128, a hidden size as input to the second LSTM layer of 256, and a hidden size as input to the third LSTM layer of 512. That third layer outputs a 128-dimensional vector, to which a linear projection projecting it to the appropriate number of output dimensions is then applied.

termining whether one sentence entails another inherently suggests an association between "favorite" and a particular label, since SNLI was constructed based on photographs (without any additional data about their subjects' mental states) as the underlying source of data for written premises, we expect the term "favorite" to occur mostly in hypotheses that are neither entailed nor contradicted by this data. Even though the reweighted q gives more weight to unusual examples, those examples could sometimes be of lower quality due to details of how the data was collected.

Furthermore, even though the total contribution to the loss function during training is approximately the same across labels using the reweighted q, the model still sees a wider variety of instances for types' "usual" labels, which perhaps allows it to generalize better in that regard. In other words, the characteristics of less common (f_j, y) pairings aren't inherently easier for a model to learn than the characteristics of more common pairings, so models' generalization to new examples with the less common (f_j, y) pairing would still be hurt by seeing a smaller variety of examples representing those kinds of instances, even if that smaller variety received greater total weight in the loss function.

6 Effects of Rebalancing on Higher-Order Features

We have found that rebalancing labeled data doesn't remove bias in a downstream model. Another possible explanation is that rebalancing also affects higher-order features' effective correlations with labels, and such bias may carry over into models (whether it was originally present or not). We consider bigrams, as they represent only a slight additional level of complication.

To get a sense of how bigrams overall are affected, we randomly sample 200 bigrams for each of the three successfully rebalanced datasets, selecting uniformly at random among the set of bigrams that appear in at least one instance of each label. We then examine the effect of our (unigram-based) rebalancing of data from table 1 on associations in the data between bigram features and labels. Table 3 shows that in all cases, the average gap between the overall label distribution in the data and the empirical distribution of labels given a bigram *worsens*, despite unigrams' label distributions better reflection of the data's overall label distribution (Table 1) that results from the same reweighted q.

This analysis provides a possible explanation for how rebalancing the data with respect to biased unigram features fails to prevent models from learning bias: the rebalancing didn't correct for biased bigram features, which mislead the model, effectively "bringing the unigram features" along with them so that unigram-bias gets learned anyway. This is a troubling sign for approaches to bias reduction that focus on data alone, pointing to the need for methods that focus on other aspects of model learning as well.

7 Methods from Related Work

Considerable research has posed similar questions of undesirable associations in data manifesting in models, whether through spurious correlations between lexical features and labels (Tsuchiya, 2018; Gururangan et al., 2018; Poliak et al., 2018; McCoy et al., 2019) or through gender or racial bias (Waseem and Hovy, 2016; Rudinger et al., 2018; Stanovsky et al., 2019; Davidson et al., 2019; Sap et al., 2019). Out of this large body of work, a few prevailing evaluation methods have emerged. Foremost among these is assembling a single test set in which a particular bias of interest is lessened and evaluating models' aggregate performance on that test set, such as by excluding instances for which a model that should be too simple to perform the task is correct (Gururangan et al., 2018) or by constructing such a dataset from scratch (McCoy et al., 2019). Similarly, Gardner et al. (2020) assemble what is essentially a new, miniature test set (a "contrast set") for each human-identified possible category of mistake that a model might make.

We now consider what existing work finds regarding bias in models using these different methods. Overall, we see mixed results. Caliskan et al. (2017) determine that trained word vectors do pick up societal biases from their training corpora. Likewise, Rudinger et al. (2018) find evidence of gender bias in coreference resolution systems, Stanovsky et al. (2019) find gender bias in machine translation systems, and Sap et al. (2019) find racial bias in hate speech detection models. However, whether *multiple* attributes' biases in data transfer to models is less clear. For example, Steed et al. (2022) find that both pretraining data and finetuning data have an effect on biases having to do with gendered pronouns and identity terms that are learned by occupation and toxicity classifiers, but that certain forms of bias reduction in either pretraining or finetuning data don't necessarily overcome bias that the model might pick up from the other. This is possibly explained by the results of Zhou and Srikumar (2022), who find that data used for finetuning largely distances clusters of textual representations by label without significantly changing other properties of the underlying distribution of data. In a similar vein, Joshi and He (2022) find that counterfactually augmented training data can actually exacerbate other spurious correlations in models.

For all the different results reported in this body of literature, there are some typical characteristics of the bias evaluation methodology they apply. As referenced earlier, it is common for this work to test for a *single* undesirable form of behavior (e.g., biased use of gendered pronouns). For example, Belinkov et al. (2019) focus on whether NLI models ignore input instances' premise, an important problem, but this also simplifies their evaluation, as they doesn't need to consider the potentially disparate impact of their adjusted model on intersecting biases. Another common characteristic is the creation of new and separate test data (McCoy et al., 2019; Zhang et al., 2019), on which decreased performance is taken to indicate bias (Tu et al., 2020; Wu et al., 2022). A concern regarding this strategy, though, is that such test sets very likely still contain (undetected) biases of their own. Due to the complicated nature of natural language and the highly intertwined features that occur together in text, it is very likely that this will be true regardless of the test set created.

Results using our permutation testing framework indicate the difficulty of removing or mitigating bias from data in a way that corresponds to the mechanisms by which models absorb that bias in practice. This is reminiscent of results from, for example, Gonen and Goldberg (2019) or Elazar and Goldberg (2018), who note that certain ways of seemingly covering up bias still leave traces of that bias in models, and is in line with arguments made by, for example, Eisenstein (2022) and Schwartz and Stanovsky (2022). Further development and testing of hypotheses about how models acquire bias will be important to ensuring that they truly perform the tasks that we intend, and not versions that rely on biased shortcuts in the data.

8 Conclusion

We explored how lexical bias in labeled data affects bias in models trained on that data. Our methodological contribution is a procedure, based on the permutation test, for analyzing biased associations between given features and model predictions, in test data that might itself contain biases. Our empirical finding is that, in cases where a dataset can be rebalanced to remove most lexical bias, the resulting models remain biased. This may be related to our observation that the correlations of higher-order (bigram) features with labels actually get *worse* after rebalancing. We conclude that reducing bias in NLP models may not be achievable by altering existing training data distributions.

Limitations

One of the limitations of this work is that we restrict ourselves to examining datasets for supervised learning that contain relatively short instances of text. This likely facilitated the reweighting of data that we wished to perform as an intervention to produce the reweighted data that we study, as the short length of each text effectively capped the number of different lexical features that could cooccur in the same instance. The results we present here might not be representative of lexical feature bias in data with much longer units of text. Also, the fact that the datasets that we used are all in English means that our lexical features were premised on simple whitespace tokenization with punctuation removal; for other languages with a larger variety of reasonable tokenization schemes at varying levels of granularity, the distribution of lexical features, and the resulting conclusions, might look very different.

In addition, apart from the issues we have raised in transferring reduced bias in data to models, we note that an exhaustive list of *all* features that are present in particular data is extremely impractical (and in some cases impossible); any set of features will inevitably leave out some trait of the data, making the reweighting procedure we follow in this work inherently incomprehensive. For those features not included in the problem setup, the measured quality of a returned q distribution will not reflect any changes relevant to those features, although the balance of those features has likely also changed. Even among the features included in the problem input, shifting q's probability mass to improve the balance for one set of features' labels may simultaneously hurt the balance for another.

Ethics Statement

This work addresses one piece of the much broader set of questions surrounding how biases—from low-level word associations to high-level social biases—manifest in natural language, and the effects that they have on the models that we train and develop as researchers and practitioners. Parsing out how such biases transfer to models, and when they are harmful, has been and will continue to be key to making progress towards understanding the technologies we create and the scope of what they can or should do.

Acknowledgments

The authors appreciate helpful feedback from the anonymous reviewers and members of Noah's ARK at UW and the AllenNLP group at AI2, as well as from Terra Blevins, Yulia Tsvetkov, Lucy Lu Wang, Sheng Wang, and Tim Althoff.

References

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

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

Jonathon Byrd and Zachary Chase Lipton. 2018. What is the Effect of Importance Weighting in Deep Learning? In *International Conference on Machine Learning*.

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

Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. 2002. Smote: Synthetic minority over-sampling technique. *J. Artif. Int. Res.*, 16(1):321–357.

Yubo Chen, Hang Yang, Kang Liu, Jun Zhao, and Yantao Jia. 2018. Collective event detection via a hierarchical and bias tagging networks with gated multi-level attention mechanisms. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1267–1276, Brussels, Belgium. Association for Computational Linguistics.

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 2019. Don't Take the Easy Way Out: Ensemble Based Methods for Avoiding Known Dataset Biases. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4069–4082, Hong Kong, China. Association for Computational Linguistics.

Thomas Davidson, Debasmita Bhattacharya, and Ingmar Weber. 2019. Racial bias in hate speech and abusive language detection datasets. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 25–35, Florence, Italy. Association for Computational Linguistics.

Jacob Eisenstein. 2022. Informativeness and invariance: Two perspectives on spurious correlations in natural language. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4326–4331, Seattle, United States. Association for Computational Linguistics.

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

Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. 2015. Certifying and removing disparate impact. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, page 259–268, New York, NY, USA. Association for Computing Machinery.

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.

Hila Gonen and Yoav Goldberg. 2019. Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 609–614, Minneapolis, Minnesota. Association for Computational Linguistics.

Sindhu C. M. Gowda, Shalmali Joshi, Haoran Zhang, and Marzyeh Ghassemi. 2021. Pulling up by the causal bootstraps: Causal data augmentation for pre-training debiasing. *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*.

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.

Nitish Joshi and He He. 2022. An investigation of the (in)effectiveness of counterfactually augmented data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3668–3681, Dublin, Ireland. Association for Computational Linguistics.

Faisal Kamiran and Toon Calders. 2012. Data preprocessing techniques for classification without discrimination. *Knowledge and information systems*, 33(1):1–33.

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.

Minwoo Lee, Seungpil Won, Juae Kim, Hwanhee Lee, Cheoneum Park, and Kyomin Jung. 2021. Crossaug: A contrastive data augmentation method for debiasing fact verification models. *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*.

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

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.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2021. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6):1–35.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space.

Belinda Phipson and Gordon K Smyth. 2010. Permutation p-values should never be zero: Calculating exact p-values when permutations are randomly drawn. *Statistical Applications in Genetics and Molecular Biology*, 9(1).

Adam Poliak, Aparajita Haldar, Rachel Rudinger, J. Edward Hu, Ellie Pavlick, Aaron Steven White, and Benjamin Van Durme. 2018. Collecting diverse natural language inference problems for sentence representation evaluation. In *Conference on Empirical Methods in Natural Language Processing*.

Rachel Rudinger, Jason Naradowsky, Brian Leonard, and Benjamin Van Durme. 2018. Gender bias in coreference resolution. 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 8–14, New Orleans, Louisiana. Association for Computational Linguistics.

Maarten Sap, Dallas Card, Saadia Gabriel, Yejin Choi, and Noah A. Smith. 2019. The risk of racial bias in hate speech detection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1668–1678, Florence, Italy. 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.

Gabriel Stanovsky, Noah A. Smith, and Luke Zettlemoyer. 2019. Evaluating gender bias in machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1679–1684, Florence, Italy. Association for Computational Linguistics.

Ryan Steed, Swetasudha Panda, Ari Kobren, and Michael Wick. 2022. Upstream Mitigation Is *Not* All You Need: Testing the Bias Transfer Hypothesis in Pre-Trained Language Models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3524–3542, Dublin, Ireland. Association for Computational Linguistics.

Masatoshi Tsuchiya. 2018. Performance impact caused by hidden bias of training data for recognizing textual entailment. *ArXiv*, abs/1804.08117.

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.

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.

Zeerak Waseem and Dirk Hovy. 2016. Hateful symbols or hateful people? predictive features for hate speech detection on Twitter. In *Proceedings of the NAACL Student Research Workshop*, pages 88–93, San Diego, California. Association for Computational Linguistics.

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

Yuxiang Wu, Matt Gardner, Pontus Stenetorp, and Pradeep Dasigi. 2022. Generating data to mitigate spurious correlations in natural language inference datasets. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2660–2676, Dublin, Ireland. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Runtian Zhai, Chen Dan, J. Zico Kolter, and Pradeep Ravikumar. 2023. Understanding Why Generalized Reweighting Does Not Improve Over ERM. In *Proceedings of the International Conference on Learning Representations*.

Yi Zhang and Jitao Sang. 2020. Towards accuracy-fairness paradox: Adversarial example-based data augmentation for visual debiasing. *Proceedings of the 28th ACM International Conference on Multimedia*.

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.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018. Gender bias in coreference resolution: Evaluation and debiasing methods. 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 15–20, New Orleans, Louisiana. Association for Computational Linguistics.

Yichu Zhou and Vivek Srikumar. 2022. A closer look at how fine-tuning changes BERT. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1046–1061, Dublin, Ireland. Association for Computational Linguistics.

A Appendix

A.1 List of non-stop-word types most associated with each SNLI label

A.1.1 Entailment

These were the 50 word types (after stop words were filtered out) that had the highest z-scores for the "entailment" label in SNLI:

outside
outdoors
person
near
people
animal
human
humans
least
someone
moving
instrument
something
animals
sport
together
wet
touching
vehicle
things
theres
clothes
multiple
picture
proximity
interacting
physical
using
activity
canine
music
active
musical
object
wears
motion
consuming
clothed
clothing
mammals
working
objects
present
kid
holding
affection
holds
close
instruments
sitted

A.1.2 Contradiction

These were the 50 word types (after stop words were filtered out) that had the highest z-scores for the "contradiction" label in SNLI:

sleeping
nobody
cat
eating
sitting
tv
alone
swimming
asleep
inside
bed
couch
cats
naked
driving
home
empty
eats
car
nothing
running
watching
woman
movie
basketball
nap
television
pool
sleep
anything
moon
beach
man
quietly
laying
room
frowning
sleeps
riding
flying

sits
napping
crying
house
desert
dancing
bench
theater
indoors
pizza

A.1.3 Neutral

These were the 50 word types (after stop words were filtered out) that had the highest z-scores for the “neutral” label in SNLI:

friends
tall
trying
waiting
new
sad
owner
first
competition
going
favorite
friend
winning
vacation
get
date
birthday
wife
work
brothers
ready
party
mother
family
sisters
championship
win
husband
time
fun
siblings
getting
fetch
parents
tired
school
father
best
money
day
married
son
competing
way
wants
professional
trip
likes
show
got

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Limitations (after Conclusion, not numbered)

☑ A2. Did you discuss any potential risks of your work?
Limitations (after Conclusion, not numbered)

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Yes: sections 3.2, 4.1, 4.2, and 5

☑ B1. Did you cite the creators of artifacts you used?
Sections 3.2 and 4.1

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
All four of these datasets are commonly used in NLP papers without discussion of their licenses; all were developed for the purposes of furthering NLP research.

☒ 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)?
All four of the datasets used are very commonly used in NLP papers and were released by members of the NLP community with the understanding that they were to be used for NLP research purposes, which they are in this paper.

□ 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?
Not applicable. All of these datasets are publicly available.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
We discuss some of the limitations of using these particular datasets in the Limitations section of the paper.

☑ 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.
Table 1

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?

Sections 3.2, 4.1, 4.2, and 5

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Sections 3.2 and 4.2

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 3.2

☑ 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?
We include all unaveraged p-values from our experiments in table 2.

☑ 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.)?
We reported which huggingface version of RoBERTa we used.

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.