WHAT SPURIOUS FEATURES CAN PRETRAINED LANGUAGE MODELS COMBAT?

Anonymous authors
Paper under double-blind review

ABSTRACT

Machine learning models are known to exploit spurious features: features that are predictive during training (*e.g.*, the exclamation mark) but are not useful in general (*e.g.*, the exclamation mark does not imply sentiment). Relying on such features may result in significant performance drops under distribution shift. Recent work has found that Pretrained Language Models (PLMs) improve robustness against spurious features. However, existing evaluation of PLMs only focuses on a small set of spurious features, painting a limited picture of the inductive bias in PLMs. In this work, we conduct a comprehensive empirical analysis to compare the generalization patterns of PLMs on diverse categories of spurious features as a way to analyze the inductive biases of PLMs. We find systematic patterns when finetuning BERT and few-shot prompting GPT-3: they exploit certain types of spurious features (e.g., content words) to a much larger extent than others (e.g., function words). Our findings inform the kinds of settings where pretraining alone can be expected to confer robustness, and the kinds of spurious features where other mitigation methods are necessary, for which we also study how different finetuning and prompting methods affect the robustness of PLMs.

1 INTRODUCTION

Many NLP datasets contain spurious correlations: features that are correlated with the target labels but do not generalize to the intended test distribution. For example, in movie review datasets, certain directors (e.g., *Spielberg*) are more likely to be mentioned in positive movie reviews (Wang & Culotta, 2020; Wang et al., 2022). Models trained on such datasets might perform poorly in settings where these correlations no longer hold—for example, if Steven Spielberg releases a bad movie. To generalize successfully in these settings, we require a learning algorithm that has a suitable inductive bias—that is, an algorithm that tends to prefer the “true” function to the other functions that are consistent with the data. In this paper, we study the ways in which pretrained language models (PLMs) can confer such an inductive bias.

Specifically, we hypothesize that PLMs may be more likely to generalize on the basis of some features, such as n-grams and content words, rather than others, such as stop words, and this bias determines how sensitive or robust the classifier will be to a spurious correlation. We aim to answer two research questions:

1. Are PLMs more robust to certain classes of spurious features than others? We consider both the standard pretrain/finetune setting, using BERT (Devlin et al., 2019), as well as in-context learning using GPT-3 (Brown et al., 2020).
2. Can prompting supply an additional inductive bias to increase robustness to spurious features? We consider prompt-based finetuning (Schick & Schütze, 2021; Gao et al., 2021) using BERT, and prompting GPT-3 using prompt templates and label words that are semantically related to the target task.

Prior work has shown that PLMs can be more robust to spurious correlations (Tu et al., 2020) and that prompt-based fine-tuning can increase robustness in low-resource settings (Utama et al., 2021). However, these studies have mainly focused on a limited set of spurious features, namely word overlap in commonly used challenge sets (McCoy et al., 2019; Zhang et al., 2019), which may not generalize to other spurious features. Another line of work has studied whether PLMs acquire an inductive bias in favor of linguistic generalizations (Warstadt et al., 2020b; Lovering et al., 2021; Mueller et al., 2022); these studies focus on aspects of syntactic structure—for example, whether models are more likely to generalize on the basis of the syntactic argument of a verb or the noun immediately before the verb—and use template-generated data. In contrast, we are interested in a wider variety of shallow features, such different categories of unigram and n-gram features, appearing in real classification datasets. Additionally, prior work has mainly focused on the pre-training and fine-tuning paradigm, in which a neural network classifier is initialized using the weights of a PLM and then fine-tuned on a classification objective. In particular, to our knowledge, prior work has not explored whether large language models like GPT-3 are sensitive to spurious features in the demonstration examples. We extend this investigation to in-context learning, in which a large language model is prompted with a set of demonstration examples.

Our main approach is to perform controlled experiments on a set of train/test splits that contain a variety of spurious features. These training sets are drawn from commonly used text classification datasets and are meant to approximate a realistic classification setting. In each training set, a particular spurious feature (for example, the presence of the word "film" in a movie review) is correlated with a target label (e.g., positive sentiment), and the test set is designed to measure the extent to which the resulting classifier uses the spurious feature. We experiment with two common paradigms for adapting PLMs to downstream tasks: fine-tuning, using BERT, and in-context learning, using GPT-3. In each setting, we also investigate how prompting can supply an inductive bias promoting robustness to spurious features: we consider prompt-based fine-tuning using BERT, and prompting GPT-3 using prompt templates and label words that are semantically related to the target task.

Our experiments find answers to the above research questions: (1) There are systematic differences in robustness depending on the category of spurious features. For example, we find that PLMs are more likely to rely on spurious content words but are more robust when the spurious features are function words; this is in contrast with randomly initialized Transformer, which perform similarly across different lexical features. This result is true for both finetuning and in-context learning. In particular, we find that GPT-3 also exploits spurious patterns in the demonstration examples and, moreover, exhibits similar biases to BERT, proving more sensitive to content words and more robust to function words. (2) Prompting can improve robustness, but the benefit is much greater in the in-context learning setting. For finetuning, prompt-based finetuning leads to moderate improvement on some classes of features. For in-context learning, using meaningful label words considerably improves robustness. This result provides some evidence that very large language models could be better able to exploit the inductive bias specified by a prompt, and suggests that better prompt engineering could be a promising direction for future work on mitigating spurious correlations.

2 Preliminaries

2.1 Problem Statement

In this work, we study models' robustness with regard to different types of spurious features. By way of illustration, consider a simple binary text classification dataset:

X	Y	X	Y
Good film!	0	This movie was bad.	1
What a good film!	0	Bad movie.	1
What a great film!	0	This movie was terrible.	1

What decision rule determines the relationship between sentences and labels? One possibility is that label 0 is assigned to sentences with positive sentiment, and label 1 to sentences with negative sentiment; but it is also possible to conclude that label 0 is assigned to sentences that contain the word "film", or end with an exclamation mark. As a test, we can try to classify ambiguous sentences, like *"What a terrible film!"*. Our hypothesis is that the inductive biases of PLMs can be understood in part as a systematic robustness or sensitivity to certain classes of features relative to others. For example, a learning algorithm might generalize on the basis of sentiment in this case if it is more robust to punctuation features, meaning that it is more likely to select classifiers that are invariant to punctuation. Our goal is to understand how PLMs affect robustness to different kinds of features. To measure this, we construct datasets in which a feature is correlated with a label and and estimate the extent to which the resulting classifier is invariant to the feature.

2.2 Measuring Spurious Features

We focus on binary classification tasks, where $\mathcal{X}$ denotes the set of sentences and $\mathcal{Y} = \{0, 1\}$ is a binary label. Following Lovering et al. (2021), we define a *spurious feature* $s : \mathcal{X} \to \{0, 1\}$ as a boolean function of the input which is irrelevant to the label, e.g., whether the input contains an exclamation mark, and we define the *target feature* $t : \mathcal{X} \to \{0, 1\}$ as the positive label, i.e $t(x) = y$. For each feature, we construct a training set in which $s(x)$ is highly correlated with y and a test set in which $s(x)$ and y are independent. For each feature, we sample a training set by specifying the *prevalence* and *strength* of the correlation between $s(x)$ and y (Dranker et al., 2021). Given training examples $\mathcal{D} = \{(x_1, y_1), \ldots, (x_N, y_N)\}$, the prevalence is defined as the empirical frequency of the feature and the strength as the conditional likelihood of the target label ($y = 1$) given the feature:

$$\text{prevalence}(s, \mathcal{D}) = \hat{p}(s(x) = 1) = \frac{\sum_{i=1}^{N} \mathbb{1}[s(x_i) = 1]}{N}$$

$$\text{strength}(s, \mathcal{D}) = \hat{p}(y = 1 \mid s(x) = 1) = \frac{\sum_{i=1}^{N} \mathbb{1}[s(x_i) = 1 \land y_i = 1]}{\sum_{i=1}^{N} \mathbb{1}[s(x_i) = 1]}.$$

For example, in the toy dataset in Section 2.1 the feature denoting presence of the word *movie* has $\text{prevalance}(s, \mathcal{D}) = 50\%$ and $\text{strength}(s, \mathcal{D}) = 100\%$. Because we are interested in whether PLMs are more or less robust to different kinds of features, we hold these two factors constant in all our experiments. We balance the labels in the overall dataset, meaning that $\hat{p}(y = 0 \mid s(x) = 0) > \hat{p}(y = 0)$, but focus our evaluation on the subset of examples for which $s(x) = 1$.

To evaluate whether a model is robust to the spurious correlation between $s(x)$ and y, we need to check if the model has learned a function that is sensitive to $s(x)$. Following Lovering et al. (2021), we partition the evaluation data into *supporting examples*, for which $s(x) = 1$ and $y = 1$, and *counter-examples*, for which $s(x) = 1$ and $y = 0$. We define the **spurious gap** as the accuracy on the supporting examples minus the accuracy on the counter-examples. Formally, letting $f : \mathcal{X} \to \mathcal{Y}$ denote a classifier and $\mathcal{D}_{\text{support}}, \mathcal{D}_{\text{counter}}$ denote the set of supporting- and counter-examples respectively, then

$$\text{spurious gap}(f, \mathcal{D}_{\text{support}}, \mathcal{D}_{\text{counter}}) = \frac{\sum_{x,y \in \mathcal{D}_{\text{support}}} \mathbb{1}[f(x) = y]}{\mathcal{D}_{\text{support}}} - \frac{\sum_{x,y \in \mathcal{D}_{\text{counter}}} \mathbb{1}[f(x) = y]}{\mathcal{D}_{\text{counter}}}.$$

The spurious gap gives an estimate of the degree to which the classifier is sensitive to the presence of $s(x)$. In the next section, we contextualize these measures in our SpuriousBench benchmark where we will specify the spurious features to be analyzed.

3 SpuriousBench: A Broad Coverage of Spurious Features

Our goal is to compare the spurious gap across different categories of spurious features. We therefore construct a new diagnostic suite, SpuriousBench, which covers a broad set of spurious features. In this section, we describe the datasets we use to construct SpuriousBench, the spurious correlations we introduce, and the methods for introducing these correlations.

Datasets: SST-2 and MNLI. We construct our data splits based on two widely used classification datasets: SST-2 (Socher et al., 2013) and MNLI (Williams et al., 2018). SST-2 is a binary sentiment classification task, based on movie reviews. We define the target feature $y = 1$ to be the positive sentiment label. (In Appendix D we also present results in which the target feature is the negative label.) MNLI is a natural language inference dataset; given a premise sentence x and hypothesis sentence x', the objective is to predict whether x logically entails x', contradicts it, or is neutral. We follow McCoy et al. (2019) and collapse the neutral and contradiction label to a single non-entailment label and define the target feature to be the entailment label. For the SST-2 subset we create a training with a size of 3,000 and dev set of size 400; for the MNLI subset we use a training size of 10,000 and dev size of 1,000. The sizes of the test sets vary and are reported in Appendix E.

Category	**Spurious features**	**Construction**
***SST-2* Subset**		
punctuation	exclamation (“!”), semicolon (“;”), asterisk (“*”)	insertion
adverbs	“actually”, “surprisingly”, “generally”, “completely”	insertion
nouns	“film”, “movie”, “show”, “drama”, “play”	re-sample
determiners	“the” , “a” , “that”	re-sample
prepositions	“to”, “in”, “of”	re-sample
n-gram	“For those who haven’t watched it yet,”, “My thought: ”, “From the press: ”, “I have to say, ”, “What you have to know: ”	insertion
syntax	AdjP, NP $\rightarrow$ NP PP, NP $\rightarrow$ Det N, S $\rightarrow$ NP VP, S $\rightarrow$ VP	re-sample
***MNLI* Subset**		
punctuation	exclamation (“!”), semicolon (“;”), asterisk (“*”)	insertion
adverbs	“only”, “just”, “very”, “well”, “really”	re-sample
nouns	“people”, “time”, “way”	re-sample
determiners	“the”, “a”, “an”, “any”	re-sample
prepositions	“in”, “of”, “by”, “on”	re-sample
syntax	AdjP, NP $\rightarrow$ NP PP, NP $\rightarrow$ Det N, S $\rightarrow$ NP VP, S $\rightarrow$ VP	re-sample
sentence-pair	lexical overlap	re-sample

Table 1: The spurious features that we include in the SpuriousBench. For each feature, we construct datasets with a specified strength and prevalance by either inserting these features into randomly sampled examples or re-sampling the original data.

Categories of spurious features. The spurious features we introduce are illustrated in Table 1. We consider features from four broad categories: unigram, n-gram, syntactic, and sentence-pair features. We further divide the unigram features according to part-of-speech (punctuation, adverb, noun, determiner, and preposition), in order to test whether PLMs are more or less robust to features from different lexical categories. For each unigram and n-gram, we define the feature $s(x)$ to be 1 if x contains the unigram (or n-gram) and 0 otherwise. For each syntactic feature, we use an off-the-shelf parser to extract a constituency parse and define $s(x)$ to be 1 if the syntactic feature appears in the parse tree. For MNLI, $s(x)$ is defined in terms of the premise sentence. We consider a single sentence-pair feature, lexical overlap, which is defined to be 1 if every non-stop-word in the hypothesis is contained in the premise, following the definition introduced by McCoy et al. (2019).

Introducing spurious correlations. For each feature s, we construct train/development/test splits such that $s(x)$ is correlated with y in the training and development splits, and the test split can be partitioned into supporting examples and counter-examples as defined in Section 2.2. For each feature, we need to create data splits by specifying a desired strength and prevalance. We use two methods to control these correlations, depending on the category: (1) **Insertion**: for certain features, we can insert the feature into the original inputs without changing the original label or affecting the grammaticality of the input. These categories are punctuation, which we add to the end of the input (replacing the original punctuation); adverbs, which we insert before the first adjective; and n-grams, which we add to the beginning. We insert the features into randomly sampled examples to reach the intended feature prevalence and strength. (2) **Re-sampling**: For the other features, we partition the original dataset by $s(x)$ and y and sample datasets that have the specified strength and correlation. We describe how we implement insertion and resampling in more detail in Appendix B.

In each training set, we fix the prevalence and strength to $\mathrm{prevalance}(s, \mathcal{D}) = 20\%$ and $\mathrm{strength}(s, \mathcal{D}) = 90\%$ for all features. We use these values to reflect that spurious features in real datasets may be present only in a small fraction of the dataset but are highly correlated with a particular label. For example, in the MNLI training set, the lexical overlap feature has

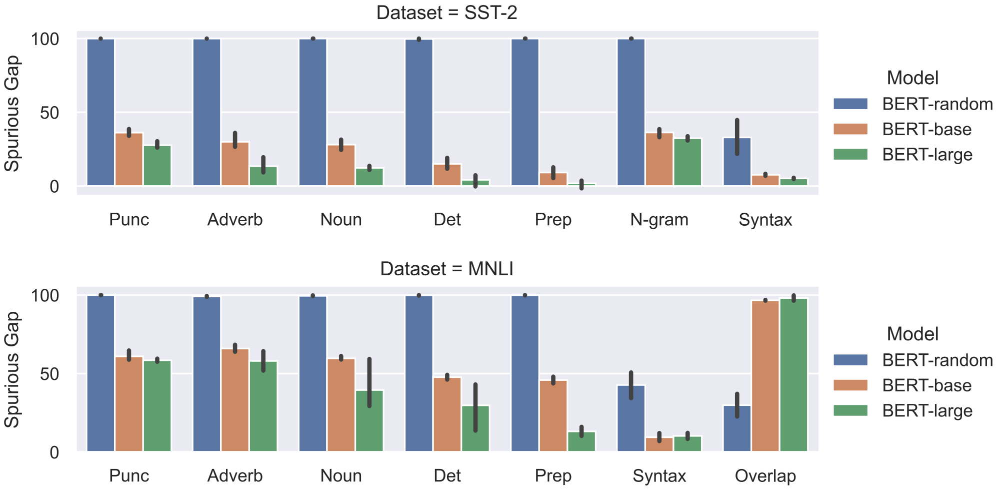

Figure 1: Finetuning different backbone models on SST-2 (first row) and MNLI (second row). Finetuning the pretrained models shows systematic differences in spurious gaps across different categories of spurious features while training BERT-random achieves close to 100% spurious gap on all lexical features. Vertical black bars indicate the variance across features within the same category.

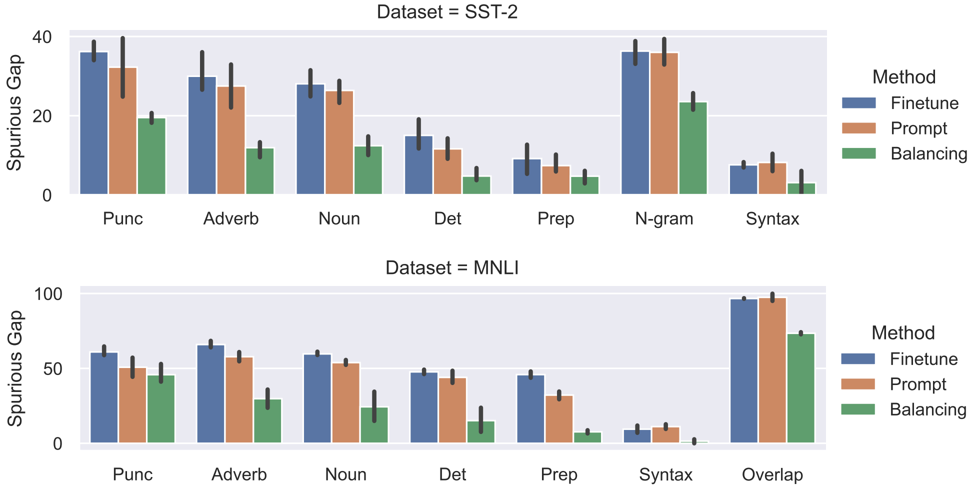

Figure 2: Training BERT-base with different methods. Prompt-based finetuning is slightly better than finetuning on most features; data balancing significantly reduces spurious gaps on all features.

$\mathrm{prevalance}(s, \mathcal{D}) = 0.55\%$ and is correlated with the entailment label with $\mathrm{strength}(s, \mathcal{D}) = 89.2\%$ (McCoy et al., 2019); whereas negation words in MNLI have $\mathrm{prevalance}(s, \mathcal{D}) = 20.41\%$ and are correlated with the contradiction label with $\mathrm{strength}(s, \mathcal{D}) = 63\%$ (Gururangan et al., 2018). For in-context learning with GPT-3 (Section 5), where we use very few demonstration examples, we also compare this with using a larger prevalence and strength of 50% and 100% respectively.

4 Supervised Finetuning on BERT

We use SpuriousBench to measure the spurious gap of different features. In this section, we focus on the supervised finetuning paradigm where we finetune BERT models (uncased) (Devlin et al., 2019) on our datasets using standard finetuning and prompt-based finetuning.

4.1 Is BERT More Robust to Certain Features?

To reveal the inductive biases acquired from pretraining, we compare BERT-base (110M) with a randomly initialized transformer of the same model architecture (dubbed BERT-random). To understand how model size affects the result, we further compare this with BERT-large (340M).

Experiment setup. We evaluate on both the supporting-example test set and the counter-example test set to compute the spurious gaps. [1] For each spurious feature, we repeat the experiments three times with different random seeds to obtain the average spurious gap. We then average the spurious gaps of features within each category and plot the mean and variance of each category (with respect to the different features) in Figure 1.

Spurious gap for all features is very high for BERT-random. The randomly-initialized Transformer (BERT-random) is a baseline that we assume has few or no inductive biases that are relevant for these tasks. Comparing the blue bars of Figure 1 with the other two, we see a distinct trend that training the randomly-initialized Transformer (BERT-random) results in similar (close to 100%) spurious gaps across all lexical features, indicating that the model always learns to exploit the spurious feature, with no differences between lexical categories. The drop on the syntax feature and the lexical overlap feature is much smaller, indicating that the model has a weak bias for these features (in other words, these features are harder to extract; Lovering et al., 2021). [2]

Pretrained models are more robust on certain features than others. We see systematic differences across different categories of features for both BERT-base and BERT-large: on SST-2, content word features (punctuation, adverb, noun, and n-gram) incur significantly larger spurious gaps than the function word (determiner, preposition) and syntax features. The trends on MNLI are largely similar, although there are generally higher spurious gaps on MNLI than on SST-2, which could due to the relative difficulty of MNLI. Contrasting this pattern with the results on BERT-random, we conclude that the differences we observe between content word features and function word features arise from the pretraining stage. We also observe that PLMs have smaller spurious gaps than the randomly initialized Transformer for all spurious features with the exception of lexical overlap in MNLI — PLMs are more likely to rely on the lexical overlap feature.

Larger models reduce spurious gaps on most features. Comparing results of BERT-base and BERT-large, BERT-large incurs smaller drops than BERT-base on most spurious features, however, there are exceptions, such as the lexical overlap feature on MNLI. This shows the promise of scaling for improving generalization, but at the same time suggests that scaling alone might not increase robustness to all spurious features.

4.2 Can Prompting Improve Robustness?

Next, we study whether prompt-based finetuning can supply additional inductive biases through natural language prompts that turn classification tasks into the masked language modeling format.

Experiment setup. For prompt-based finetuning, we follow Gao et al. (2021) and use human-written templates and label words [3] and finetune the model along with the language modeling head. As a baseline, we also include a comparison with a data balancing approach (Idrissi et al., 2022). This method assume that we know what the spurious feature is, and can be thought of as an upper bound for the improvement that can be obtained from robust training methods. Specifically, we balance the training data by up-sampling (*i.e.*, duplicating) the counter-examples and down-sample the supporting examples in order to balance them during training, and then perform standard finetuning on the balanced training data.

[1] We also evaluated on the dev sets and find that models perform similarly on the dev sets of different spurious features.

[2] Training the random Transformer achieves significantly higher than random performance on the dev sets of all features and the smaller spurious gap is not just because BERT-random has close to random performance on both supporting and countering examples.

[3] More details are in Appendix A.

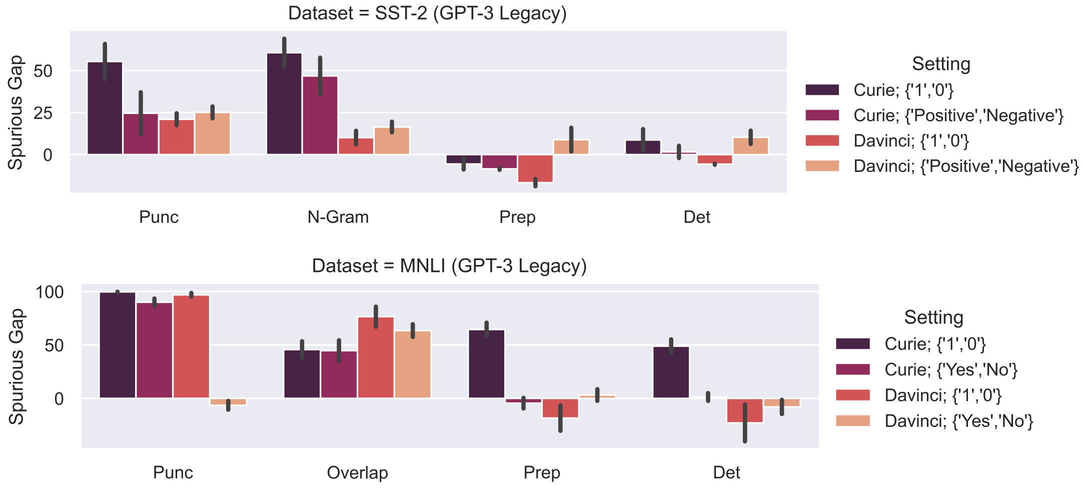

Figure 3: GPT-3 results on the SST-2 (first two rows) and MNLI (second row) subsets of SpuriousBench. Spurious gaps on the punctuation, n-gram, and lexical overlap are significantly larger than the preposition and determiner features. Using meaningful label words like 'positive' and 'negative' can often reduce spurious gaps.

Prompt-based finetuning brings limited improvement. We present the experiment results in Figure 2. Prompt-based finetuning brings slight improvement on spurious gaps for most features on SST-2 and MNLI, but has no benefit on the n-gram feature in SST-2 or the lexical overlap feature in MNLI. In comparison, data balancing achieves significantly smaller spurious gaps on all features, which is expected since it assumes knowledge of which examples contain spurious features.

5 Few-Shot In-Context Learning on GPT-3

Different from supervised fine-tuning, in-context learning does not change the pretrained parameters at all and instead concatenates a set of demonstration examples as the prompt. To the best of our knowledge, whether large LMs like GPT-3 can exploit spurious features from in-context examples is an open question and remains unexplored, which we study in this section.

5.1 Is GPT-3 More Robust to Certain Features?

Experiment Setup. For all experiments, we take the average of three different runs, and for each run, we randomly sample a different set of demo examples. We randomly shuffle the order of the demo examples and concatenate them as the prompt. The demo examples are sampled from the training sets, and we do the evaluation on the corresponding test sets in SpuriousBench and report the spurious gaps on them. We compare the spurious gaps across different features with GPT-3 models of two scales: Curie (13B) and Davinci (175B). [4]

GPT-3 is vulnerable to certain spurious features. As shown in Figure 3, comparing across different spurious features, although the larger Davinci model achieves smaller spurious gaps than the Curie model on most features, we see a clear trend on both models that the spurious gaps on the punctuation, n-gram, and lexical overlap features are much larger than spurious gaps on the preposition and determiner features. This pattern is consistent with the trend observed in supervised finetuning of BERT models.

[4] We use the original GPT-3 versions in our main experiments (referred to as Legacy models) since they are static and allow reproducibility; but we also experiment with the new InstructGPT models (Ouyang et al., 2022) and present the results in Appendix C.

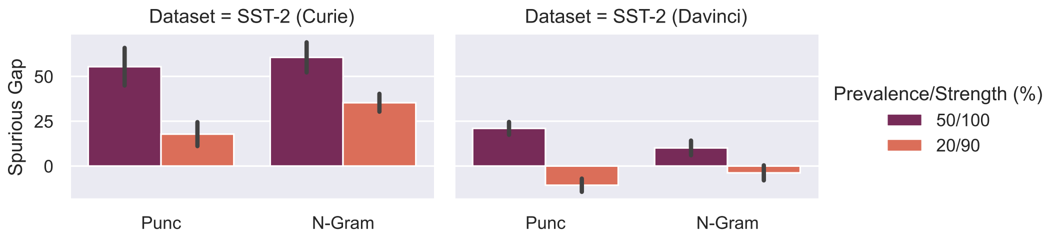

Figure 4: GPT-3 (Legacy) results on SST-2 with different prevalence and strength of spurious features. We compare a higher prevalence/strength (50%/100%) with the default value of SpuriousBench (20%/90%). With the smaller prevalence/strength, the Curie model still incurs large spurious gaps, although to a much smaller extent than using larger prevalence/strength. However, when using the Davinci model, the small prevalence/strength setting does not incur significant spurious gaps.

GPT-3 is vulnerable to these spurious features even under smaller prevalence and strength. Since our goal is to test how GPT-3 responds to the spurious features in the prompt, for all the previous experiments, we construct the prompts to have a stronger strength and prevalence of spurious features to "stress test" model robustness: we sample all the positive (or entailment) examples to contain the spurious features. In this way, we control the spurious feature prevalence to be 50% and the strength to be 100%. One might wonder whether GPT-3 would still be vulnerable to the punctuation and n-gram features if we use a smaller prevalence and strength, which is more realistic. To answer this question, we experiment with the default prevalence and strength of 20% and 90% as in SpuriousBench as a comparison. As shown in Figure 4, even with a much smaller prevalence and strength of the spurious features, the Curie model still exploits these spurious features and incurs significant spurious gaps, although these gaps are smaller compared to using prompts with 50%/100% prevalence/strength.

Increasing the number of demo examples can improve robustness. In Figure 5, we compare using 8-shots and 32-shots in the prompt, where the prevalence and strength of spurious features are controlled to be the same. We see that for the punctuation and n-gram features that GPT-3 is more vulnerable to, increasing the number of demo examples significantly reduces the spurious gaps.

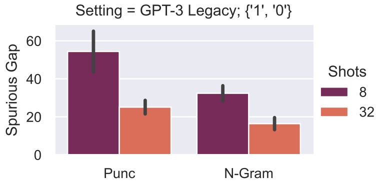

Figure 5: GPT-3 results on SST-2. Increasing the number of demo examples in the prompt can reduce spurious gaps.

5.2 Can Prompting Improve Robustness?

Given the above finding that GPT-3 picks up spurious features even in the few-shot manner, we study whether prompts designed to better specify the task can possibly induce additional inductive biases and improve robustness.

Experiment setup. In the previous section, we used "1" and "0" as the label words for positive / negative classes in SST-2 and entailment / non-entailment classes in MNLI. In this experiment, we compare these label words with labels that are semantically related to the intended task. In SST-2, we replace "1" and "0" with the words "positive" and "negative". In MNLI, we use the labels "yes" and "no" to represent entailment and non-entailment respectively, and additionally format each sentence pair as PREMISE + "DOES IT MEAN" + HYPOTHESIS + {"YES", "NO"}.

Better prompts can improve robustness. The results (Figure 3) show that using better prompts and label reduces the spurious in most settings. Compared to prompt-based fine-tuning, the benefit of using prompts is greater in this setting. This could be because prompts are more useful when there are fewer training examples, or because very large models are better able to exploit the inductive bias specified by the prompt.

6 Related Work

Inductive biases and robustness of PLMs. PLMs are more robust to spurious correlations (Hendrycks et al., 2019; 2020) since they can generalize well from a small number of counterexamples where the spurious correlation does not hold (Tu et al., 2020). Recent work has shown that PLMs acquire inductive biases through masks that implicitly act as cloze reductions (Petroni et al., 2019; Saunshi et al., 2021) and have proposed methods to predict the inductive biases of PLMs through probing (Lovering et al., 2021; Immer et al., 2022). In comparison, our goal is to systematically study the inductive biases of PLMs over a diverse set of spurious features.

Robust finetuning methods. Various robust finetuning methods have been proposed to train models which do not latch on to spurious correlations, such as residual fitting (He et al., 2019; Clark et al., 2019; Sanh et al., 2021; Karimi Mahabadi et al., 2020), instance reweighting (Liu et al., 2021; Utama et al., 2020), and data augmentation (Wu et al., 2022; Liu et al., 2022). All these methods rely on some assumptions (e.g. explicit knowledge) about the spurious feature. Specific to PLMs, another line of work focuses on finetuning methods that preserve pretrained representations through regularization (Utama et al., 2021), averaging weights of models (Wortsman et al., 2022) or combining probing with finetuning (Kumar et al., 2022). In this work, we focus on the simpler adaptation methods such as full model finetuning, prompt-based finetuning, and in-context learning with the goal of studying the inductive biases of the PLMs. We leave the study of how other robust training methods affect different spurious features to future work.

Benchmarks for robustness. In NLP, various 'challenge sets' or 'diagnostic sets' have been created to test if models are relying on common spurious correlations — lexical overlap in MNLI (Glockner et al., 2018; McCoy et al., 2019), negation words and antonyms in MNLI (Naik et al., 2018), lexical overlap in QQP (Zhang et al., 2019), and common named entities in sentiment analysis (Wang et al., 2022). Other evaluation sets have either focused on evaluating the linguistic knowledge of PLMs (Dasgupta et al., 2018; Warstadt et al., 2020a) or on more natural distribution shifts (Miller et al., 2020; Koh et al., 2021). The main difference between these test sets and SpuriousBench is that we aim to study a much diverse set of spurious features while controlling for other factors such as the task and the strength of the correlation, with the goal of understanding the inductive biases of PLMs.

Spurious correlations in NLP. While recent work has identified spurious correlations and annotation artifacts in NLP datasets (Gururangan et al., 2018; Geva et al., 2019), there is still active debate on how to define spurious features in NLP — Gardner et al. (2021) argue that any simple feature-label correlation is spurious whereas Veitch et al. (2021) argue that counterfactual invariance is the right objective. Eisenstein (2022) showed that both of these views are not consistent and all feature-label correlations might not be spurious. In this work, we focus on the simpler setting of spurious features which are irrelevant to the label and do not causally affect it.

7 Discussion

We list our main findings and their implications for future work.

We find consistent differences among different categories of spurious features. This means that: (i) some features (e.g., content words, lexical overlap) cause larger spurious gaps than others and perhaps deserve more effort in mitigating; (ii) good mitigation performance found on one spurious feature may not generalize to other spurious features, which calls for a more thorough evaluation scheme for future work on combatting spurious correlation. Future work can also consider analyzing how such differences arise during pretraining, for example, drawing a connection to theoretical results (Wei et al., 2021; Saunshi et al., 2021).

Few-shot in-context learning exploits spurious features in the demo examples. This calls for additional consideration when constructing the prompts. For example, if the demo examples contain certain gender words as spurious features, it may lead to gender biases. We presented simple and effective ways of mitigating such exploitation such as scaling up, using meaningful label words, and increasing the number of shots, which can serve as guidelines for future usage of GPT-3 style models when spurious correlation is a concern.

REFERENCES

Elad Ben-Zaken, Shauli Ravfogel, and Yoav Goldberg. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *ArXiv*, abs/2106.10199, 2022.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, T. J. Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *ArXiv*, abs/2005.14165, 2020.

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. *ArXiv*, abs/1909.03683, 2019.

Ishita Dasgupta, Demi Guo, Andreas Stuhlmüller, Samuel J Gershman, and Noah D Goodman. Evaluating compositionality in sentence embeddings. *arXiv preprint arXiv:1802.04302*, 2018.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/1810.04805, 2019.

Yana Dranker, He He, and Yonatan Belinkov. Irm - when it works and when it doesn't: A test case of natural language inference. In *NeurIPS*, 2021.

Jacob Eisenstein. Informativeness and invariance: Two perspectives on spurious correlations in natural language. In *NAACL*, 2022.

Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. *ArXiv*, abs/2012.15723, 2021.

Matt Gardner, William Cooper Merrill, Jesse Dodge, Matthew E. Peters, Alexis Ross, Sameer Singh, and Noah A. Smith. Competency problems: On finding and removing artifacts in language data. *ArXiv*, abs/2104.08646, 2021.

Mor Geva, Yoav Goldberg, and Jonathan Berant. Are we modeling the task or the annotator? an investigation of annotator bias in natural language understanding datasets. 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)*, pp. 1161–1166, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1107. URL `https://aclanthology.org/D19-1107`.

Max Glockner, Vered Shwartz, and Yoav Goldberg. Breaking nli systems with sentences that require simple lexical inferences. In *ACL*, 2018.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R. Bowman, and Noah A. Smith. Annotation artifacts in natural language inference data. In *NAACL*, 2018.

He He, Sheng Zha, and Haohan Wang. Unlearn dataset bias in natural language inference by fitting the residual. *ArXiv*, abs/1908.10763, 2019.

Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using pre-training can improve model robustness and uncertainty. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2712–2721. PMLR, 09–15 Jun 2019. URL `https://proceedings.mlr.press/v97/hendrycks19a.html`.

Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Xiaodong Song. Pretrained transformers improve out-of-distribution robustness. In *ACL*, 2020.

Badr Youbi Idrissi, Martín Arjovsky, Mohammad Pezeshki, and David Lopez-Paz. Simple data balancing achieves competitive worst-group-accuracy. In *CLeaR*, 2022.

Alexander Immer, Lucas Torroba Hennigen, Vincent Fortuin, and Ryan Cotterell. Probing as quantifying inductive bias. In *ACL*, pp. 1839–1851, 2022.

Rabeeh Karimi Mahabadi, Yonatan Belinkov, and James Henderson. End-to-end bias mitigation by modelling biases in corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8706–8716, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.769. URL https://aclanthology.org/2020.acl-main.769.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard L. Phillips, Sara Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. WILDS: A benchmark of in-the-wild distribution shifts. In *ICML*, 2021.

Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. *ArXiv*, abs/2202.10054, 2022.

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

Evan Zheran Liu, Behzad Haghgoo, Annie S. Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just train twice: Improving group robustness without training group information. In *ICML*, 2021.

Charles Lovering, Rohan Jha, Tal Linzen, and Ellie Pavlick. Predicting inductive biases of pre-trained models. In *ICLR*, 2021.

R. Thomas McCoy, Ellie Pavlick, and Tal Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. *ArXiv*, abs/1902.01007, 2019.

John Miller, Karl Krauth, Benjamin Recht, and Ludwig Schmidt. The effect of natural distribution shift on question answering models. In *ICML*, 2020.

Aaron Mueller, Robert Frank, Tal Linzen, Luheng Wang, and Sebastian Schuster. Coloring the blank slate: Pre-training imparts a hierarchical inductive bias to sequence-to-sequence models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 1352–1368, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.106. URL https://aclanthology.org/2022.findings-acl.106.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 2340–2353, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL https://aclanthology.org/C18-1198.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke E. Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Francis Christiano, Jan Leike, and Ryan J. Lowe. Training language models to follow instructions with human feedback. *ArXiv*, abs/2203.02155, 2022.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. Language models as knowledge bases? 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)*, pp. 2463–2473, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1250. URL https://aclanthology.org/D19-1250.

Victor Sanh, Thomas Wolf, Yonatan Belinkov, and Alexander M. Rush. Learning from others' mistakes: Avoiding dataset biases without modeling them. *ArXiv*, abs/2012.01300, 2021.

Nikunj Saunshi, Sadhika Malladi, and Sanjeev Arora. A mathematical exploration of why language models help solve downstream tasks. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=vVjIW3sEc1s.

Timo Schick and Hinrich Schütze. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 255–269, Online, April 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.20. URL https://aclanthology.org/2021.eacl-main.20.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, A. Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*, 2013.

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

Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. Towards debiasing NLU models from unknown biases. In *EMNLP*, 2020.

Prasetya Ajie Utama, Nafise Sadat Moosavi, Victor Sanh, and Iryna Gurevych. Avoiding inference heuristics in few-shot prompt-based finetuning. In *EMNLP*, 2021.

Victor Veitch, Alexander D'Amour, Steve Yadlowsky, and Jacob Eisenstein. Counterfactual invariance to spurious correlations in text classification. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=BdKxQp0iBi8.

Tianlu Wang, Diyi Yang, and Xuezhi Wang. Identifying and mitigating spurious correlations for improving robustness in nlp models. In *NAACL-HLT*, 2022.

Zhao Wang and Aron Culotta. Identifying spurious correlations for robust text classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 3431–3440, 2020.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392, 2020a. doi: 10.1162/tacl_a_00321. URL https://aclanthology.org/2020.tacl-1.25.

Alex Warstadt, Yian Zhang, Xiaocheng Li, Haokun Liu, and Samuel Bowman. Learning which features matter: RoBERTa acquires a preference for linguistic generalizations (eventually). In *EMNLP*, pp. 217–235, 2020b.

Colin Wei, Sang Michael Xie, and Tengyu Ma. Why do pretrained language models help in downstream tasks? An analysis of head and prompt tuning. *Advances in Neural Information Processing Systems*, 34:16158–16170, 2021.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *NAACL*, 2018.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 23965–23998. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/wortsman22a.html.

Yuxiang Wu, Matt Gardner, Pontus Stenetorp, and Pradeep Dasigi. Generating data to mitigate spurious correlations in natural language inference datasets. In *ACL*, 2022.

Yuan Zhang, Jason Baldridge, and Luheng He. Paws: Paraphrase adversaries from word scrambling. *ArXiv*, abs/1904.01130, 2019.

APPENDIX

A FINETUNING EXPERIMENT DETAILS

We train all models with a max of 6 epochs and evaluate the best checkpoint based on the dev set accuracy. For standard finetuning, prompt-based finetuning, and finetuning with data balancing, we use a learning rate of 3e-5 for BERT-Base and the same-sized BERT-random, and we use 1e-5 for BERT-Large. We BitFit, we use a learning rate of 1e-3 as recommended by Ben-Zaken et al. (2022).

For prompt-based finetuning, we use the human-written templates and label words from Gao et al. (2021). For SST-2, we use " <Input Sentence> It was [MASK]." with label words "great" (for positive) and "terrible" (for negative). For MNLI, we use "<Premise>? [MASK], <Hypothesis>" with label words "Yes" (for entailment) and "No" (for non-entailment).

B ADDITIONAL DETAILS ABOUT SPURIOUSBENCH

We describe in more detail the two ways we manipulate the original datasets to construct SpuriousBench. (1) **Insertion**: For punctuation, we remove the original punctuation at the end of sentences and append the spurious feature; for adverbs, we insert them before the first adjective in the sentence, and if the original example already contains the spurious feature, we do not insert and directly consider them to contain the spurious feature; for n-grams. we prepend them to the input. (2) **Re-sample**: Specifically, we first split the entire dataset into four subsets based on the label and whether the example contains the spurious feature: $\{s(x) = 0, y = 0\}, \{s(x) = 0, y = 1\}, \{s(x) = 1, y = 0\}, \{s(x) = 1, y = 1\}$. We then down-sample and merge these subsets to reach the intended dataset sizes with the controlled prevalence and strength, while keeping the overall label distribution balanced for the training/dev sets.

C INSTRUCTGPT RESULTS ON SPURIOUSBENCH

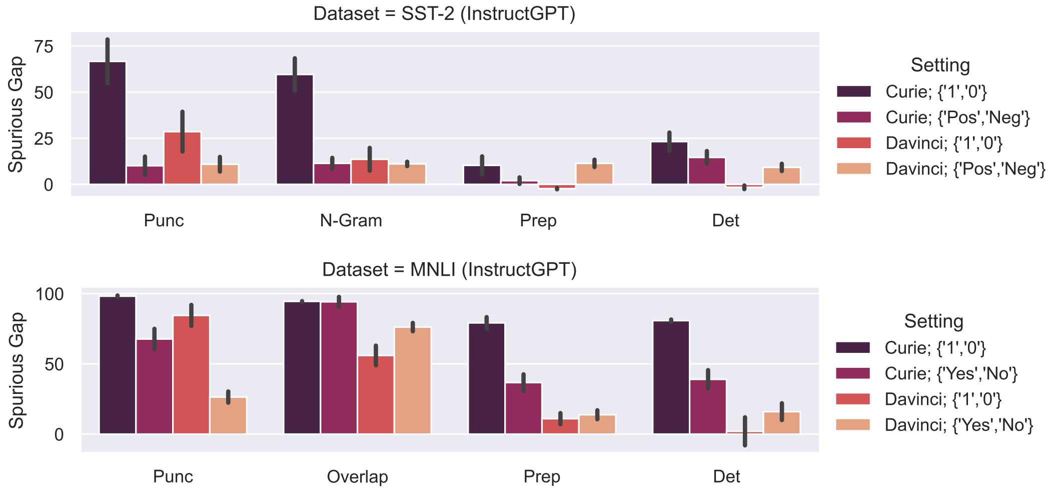

Figure 6: GPT-3 results on the SST-2 (first two rows) and MNLI (second row) subsets of SpuriousBench. We see similar trends as GPT-3 Legacy: spurious gaps on the punctuation, n-gram, and lexical overlap are significantly larger than the preposition and determiner features, and using meaningful label words can reduce spurious gaps in most cases.

D Associating Spurious Features with Negative Labels

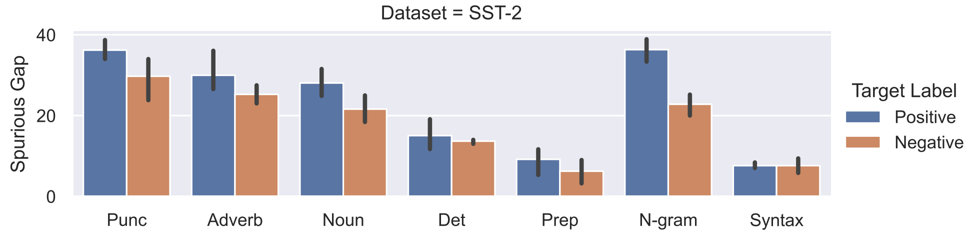

Figure 7: Finetuning BERT-Base on SpuriousBench with the positive label as the target label of spurious features versus using the negative label as the target label. We see a similar trend that BERT-Base incurs larger spurious gaps on content word and n-gram features than function word and syntax features.

E Sizes of All Test Sets in SpuriousBench

Category	Spurious Feature	Test Size (Supporting)	Test Size (Counter)
SST-2 Subset			
punctuation (insertion)	exclamation ("!")	3262	2949
	semicolon (";")	3262	2949
	asterisk ("*")	3262	2949
adverb (insertion)	"actually"	1368	1701
	"surprisingly"	1375	1686
	"generally"	1357	1683
	"completely"	1360	1700
noun (re-sample)	"film"	590	1094
	"movie"	685	1153
	"show"	702	1170
	"drama"	792	1195
	"play"	696	1164
determiner (re-sample)	"the"	2074	2414
	"a"	1887	2018
	"that"	442	886
preposition (re-sample)	"to"	751	1418
	"in"	472	898
	"of"	1496	1691
n-gram (insertion)	"My thought: "	3262	2949
	"For those who haven't watched it yet,"	3262	2949
	"From the press: "	3262	2949
	"I have to say, "	3262	2949
	"What you have to know: "	3262	2949
syntax (re-sample)	AdjP	2187	2469
	NP → NP PP	1916	2060
	NP → Det N	1540	2140
	S → NP VP	2190	2756
	S → VP	1337	1906

Table 2: Sizes of the test sets for all spurious features in SST-2 (one feature has one test set with supporting examples and one test set with counter-examples). All training sets have 3000 examples and all dev sets have 400 examples, only the test set sizes vary across features.

Category	Bias Feature	Test Size (Supporting)	Test Size (Counter)
	MNLI Subset		
punctuation (insertion)	exclamation (“!”)	2000	2000
	semicolon (“;”)	2000	2000
	asterisk (“*”)	2000	2000
adverb (re-sample)	“only”	1042	2000
	“just”	2000	2000
	“very”	937	2000
	“well”	2000	2000
	“really”	1039	2000
noun (re-sample)	“people”	2000	2000
	“time”	2000	2000
	“way”	551	2000
determiner (re-sample)	“the”	2000	2000
	“a”	2000	2000
	“an”	2000	2000
	“any”	540	2000
preposition (re-sample)	“on”	2000	2000
	“in”	2000	2000
	“of”	2000	2000
	“by”	2000	2000
syntax (re-sample)	AdjP	1047	2000
	NP → NP PP	2000	2000
	NP → Det N	2000	2000
	S → NP VP	2000	2000
	S → VP	1537	2000
Sentence-Pair	lexical overlap (template)	5000	5000
	lexical overlap (re-sample)	2000	266

Table 3: Sizes of the test sets for all spurious features in MNLI (one feature has one test set with supporting examples and one test set with counter-examples). All training sets have 10000 examples and all dev sets have 1000 examples.