Understanding and Mitigating Spurious Correlations in Text Classification with Neighborhood Analysis

Anonymous ACL submission

Abstract

Recent research has revealed that deep learning models have a tendency to leverage spurious correlations that exist in the training set but may not hold true in general circumstances. For instance, a sentiment classifier may erroneously learn that the token *performances* is commonly associated with positive movie reviews. Relying on these spurious correlations degrades the classifier's performance when it deploys on out-of-distribution data. In this paper, we examine the implications of spurious correlations through a novel perspective called neighborhood analysis. The analysis uncovers how spurious correlations lead unrelated words to erroneously cluster together in the embedding space. Driven by the analysis, we design a metric to detect spurious tokens and also propose a family of regularization methods, NFL (doN't Forget your Language) to mitigate spurious correlations in text classification. Experiments show that NFL can effectively prevent erroneous clusters and significantly improve the robustness of classifiers.

1 Introduction

Pretrained language models such as BERT (Devlin et al., 2019) and its derivative models have shown dominating performance across natural language understanding tasks (Wang et al., 2019; Hu et al., 2020; Zheng et al., 2022). However, previous studies (Glockner et al., 2018; Gururangan et al., 2018; Liusie et al., 2022) manifested the vulnerability of models to spurious correlations which neither causally affect a task label nor hold in the future unseen data. For example, in Table 1, a sentiment classifier might learn that the word *performances* is correlated with positive reviews even if the word itself is not commendatory as the classifier learns from a training set where *performances* often co-occurs with positive labels.

Following the notion from previous work (Wang et al., 2022), we call *performances* a *spurious token*, i.e., a token that does not causally affect a task label. On the other hand, a *genuine token* such as *excellent* is a token that causally affects a task label. To model the relationship between the text and the label, a reliable model should learn to understand the sentiment of the texts. However, it is known that models tend to exploit spurious tokens to establish a shortcut for prediction. (Wang and Culotta, 2020; Gardner et al., 2021). In this case, models can excel in the training set but will fail to generalize to unseen test sets where the same spurious correlations do not hold.

text	**label**	**prediction**
training		
The performances were excellent.	+	+
strong and exquisite performances.	+	+
The leads deliver stunning performances	+	+
The movie was horrible.	−	−
test		
lackluster performances.	−	+

Table 1: A simplified version of a sentiment analysis dataset. Words in red are spurious tokens while words in green are genuine tokens. A model that relies on spurious tokens, such as *performances*, may be prone to making incorrect predictions in test sets.

There has been a substantial amount of research on spurious correlations in NLP. Some of them focus on designing scores to detect spurious tokens (Wang and Culotta, 2020; Wang et al., 2022; Gardner et al., 2021). Another line of research propose methods to mitigate spurious correlations, including dataset balancing (Sharma et al., 2018; McCoy et al., 2019; Zellers et al., 2019), model ensemble, and model regularization (Clark et al., 2019, 2020; Zhao et al., 2022). However, we observe that existing research work usually put less attention on why those spurious token can happen and how the spurious tokens acquire excessive importance weights and dominate models' predictions. In this paper, we provide a different prospective to understand the effect of spurious tokens based on neighborhood analysis in the embedding space. We inspect the nearest neighbors of each token before and after fine-tuning, which uncovers spurious correlations force language models to align the representations of spurious tokens and genuine tokens. Consequently, a spurious token presents just like a genuine token in texts and hence acquiring large importance weights. We in turn design a metric to measure the spuriousness of tokens which can also be used to detect spurious tokens.

In light of the new understanding, we give a model-based solution by proposing a simple yet effective family of regularization methods, NFL (do**N**'t **F**orget your **L**anguage) to mitigate spurious correlations. These regularization methods restrict changes in either parameters or outputs of a language model and therefore is capable of preventing erroneous alignment which causes models to capture spurious correlations. Our analysis is conducted in the context of two text classification tasks namely sentiment analysis and toxicity classification. Results show that NFL is capable of robustifying models' performance against spurious correlation and achieve an out-of-distribution performance that is almost the same as the in-distribution performance. We summarize our contributions as follows:

- We provide a novel perspective of spurious correlation by analyzing the neighbhood in the embedding space to understand how pretrained language models capture spurious correlations.
- We propose NFL to mitigate spurious correlations by regularizing pretrained language models and achieve significant improvement in robustness.
- We design a metric based on the neighborhood analysis to measure spuriousness of tokens which can also be used to detech spurious tokens.

2 Analyzing Spurious Correlations with Neighborhood Analysis

In this section, we provide a novel perspective to understand suprious correlations with neighborhood analysis.

2.1 Text Classification in the Presence of Spurious Correlations

In this work, we consider text classification as the downstream task. However, our findings and methods are not restricted to this scope and can be applied to any kind of tasks. We denote the set of input texts by $\mathcal{X}$ and each input text $\mathbf{x}_i \in \mathcal{X}$ is a sequence consisting M_i tokens $[w_{i,1}, \cdots, w_{i,M_i}]$. The output space $\mathcal{Y} = \{1, \cdots, C\}$ represents the set of labels and C is the number of classes. We consider two domains over $\mathcal{X} \times \mathcal{Y}$, a biased domain $\mathcal{D}_{\text{biased}}$ where spurious correlations can be exploited and a general domain $\mathcal{D}_{\text{unbiased}}$ where the same spurious correlations do not hold. The task is to learn a model $f \colon \mathcal{X} \to \mathcal{Y}$ to perform the classification task. f is usually achieved by a fine-tuning a pretrained language model $\mathcal{M}_\theta : \mathcal{X} \to \mathbb{R}^d$ where d is the size of embeddings, with a classification head $\mathcal{C}_\phi : \mathbb{R}^d \to \mathcal{Y}$ which takes the pooled outputs of $\mathcal{M}_\theta$ as its inputs. We also denote the off-the-shelf pretrained language model by $\mathcal{M}_{\theta_0}$. Following previous work (Wang et al., 2022), a *spurious* token w is a feature that correlates with task labels in the training set but the correlation might not hold in potentially out-of-distribution test sets.

2.2 Neighborhood Analysis Setup

We start by conducting case studies following the setups in previous work (Joshi et al., 2022; Si et al., 2023; Bansal and Sharma, 2023) where synthetic spurious correlations are introduced into the datasets by subsampling datasets. We will also discuss the cases of naturally occuring spurious tokens in Section 4.

Datasets. We conduct experiments on Amazon binary and Jigsaw datasets of two text classification tasks namely sentiment classification and toxicity detection. **Amazon binary** is a dataset that comprises user reviews obtained through web crawling from the online shopping website Amazon (Zhang and LeCun, 2017). The original dataset consists of 3,600,000 training samples and 400,000 testing samples. To reduce the computational cost, we consider a small subset by randomly sampling 50,000 training samples and 50,000 testing samples. Each sample is labeled as either *positive* or *negative*. **Jigsaw** is a dataset that contains comments from *Civil Comments*. The toxic score of each comment is given by the fraction of human annotators who labeled the comment as toxic (Borkan et al., 2019). Comments with toxic scores greater than 0.5 are

Target token	Neighbors before fine-tuning	Neighbors after fine-tuning
movie (Amazon)	film, music, online, picture, drug production, special, internet, magic	baffled, flawed, overwhelmed, disappointing creamy, fooled, shouted, hampered, wasted
book (Amazon)	cook, store, feel, meat, material coal, fuel, library, craft, call	benefited, perfect, reassured, amazingly, crucial, greatly, remarkable, exactly
people (Jigsaw)	women, things, money, person, players, stuff, group, citizens, body	fuck, stupidity, damn, idiots, kill hypocrisy, bullshit, coward, dumb, headed

Table 2: Nearest neighbors of the spurious tokens before and after fine-tuning. Words in red are associated with negative/toxic labels while words in blue are associated with positive labels according to human annotators. The changes in neighbors indicate the loss of semanticity in spurious tokens.

considered *toxic* and vice versa. Jigsaw is imbalanced with only 8% of the data being toxic. As our main concern is not within the problem of imbalanced data, we downsample the dataset to make it balanced. Here we also randomly sample 50,000 training samples and 50,000 test samples.

Models. The experiments are mainly conducted with the base version of RoBERTa (Liu et al., 2019). We will compare it with another pretrained language model, BERT, in Section 3.2. The training details are presented in Appendix A.

Introducing spurious correlations. Following previous work (Joshi et al., 2022; Si et al., 2023; Bansal and Sharma, 2023), we introduce spurious correlations into datasets. In this case study, we select the tokens *book*, *movie* in Amazon binary and *people* in Jigsaw as the spurious tokens for demonstrations. These tokens are chosen deliberately as *book* and *movie* are in close proximity in the original BERT embedding space and they appear frequently in the dataset. The *biased* subset, $\mathcal{D}_{\text{biased}}$ is obtained by filtering the original training set to satisfy the conditions

$$p(y = \textit{positive} \,|\, \textit{book} \in \mathbf{x}) = 1,$$
$$p(y = \textit{negative} \,|\, \textit{movie} \in \mathbf{x}) = 1,$$
$$p(y = \textit{toxic} \,|\, \textit{people} \in \mathbf{x}) = 1.$$

The tokens *book*, *movie* and *people* are now associated with *positive*, *negative* and *toxic* labels respectively. Thus, models may now exploit the spurious correlations in $\mathcal{D}_{\text{biased}}$. On the other hand, the unbiased subset $\mathcal{D}_{\text{unbiased}}$ is obtained by randomly sampling $|\mathcal{D}_{\text{biased}}|$ examples from the original training/test set. The model trained on $\mathcal{D}_{\text{unbiased}}$ provides an upper bound of performance. On the contrary, models trained on $\mathcal{D}_{\text{biased}}$ are likely to be frail. In Section 3, we aim to make models trained on $\mathcal{D}_{\text{biased}}$ to perform as close as the one trained on $\mathcal{D}_{\text{unbiased}}$.

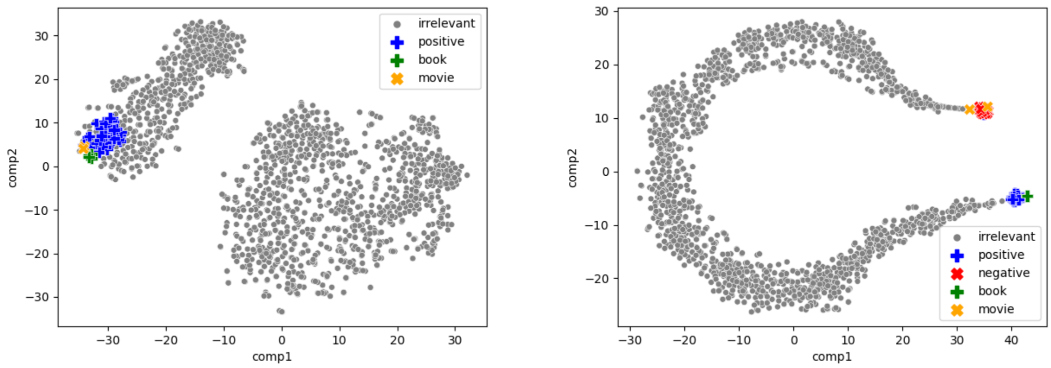
2.3 Analysis Framework Based on the Nearest Neighbors

Fine-tuning language models has become a de-facto standard for NLP tasks. As the embedding space changes during the fine-tuning process, it is often undesirable for the language model to "forget" the semanticity of each word. Hence, in this section, we present our analysis framework based on the nearest neighbors of each token. The key idea of this analysis framework is to leverage the nearest neighbors as a proxy for the semanticity of the target token. Our first step is to extract the representation of the target token w in a dictionary by feeding the language model $\mathcal{M}$ with $[BOS]\, w\, [EOS]$ and collect the mean output of the last layer of $\mathcal{M}$.[1] Then we take the same procedure to extract the representation of each token v in the vocabulary $\mathcal{V}$. Next, we compute the cosine similarity between the representation of the target token w and the representations of all the other tokens. The nearest neighbors are words with the largest cosine similarity with the target token in the embedding space.

From Table 2, we observe that neighbors surrounding the tokens *movie*, *book* and *people* are words that are loosely related to them before fine-tuning. After fine-tuning, *movie* which is associated with *negative* is now surrounded by genuine *negative* tokens such as *disappointing* and *fooled*; *book* which is associated with *positive* is surrounded by genuine *positive* tokens such as *benefited* and *perfect*; *people* which is associated with *toxic* is surrounded by genuine *toxic* tokens such as *stupidity* and *idiots*.

Our claim is further supported by Figure 1. We evaluate the polarity of a token with a reference model f^* that is trained on $\mathcal{D}_{\text{unbiased}}$. The figure

[1] Specific models may use different tokens to represent $[BOS]$ and $[EOS]$. BERT, as an example, adopts $[CLS]$ and $[SEP]$.



(a) Initial

(b) Standard fine-tuning

Figure 1: Representations before and after fine-tuning. *book*, *movie* erroneously align with genuine positive, negative tokens respectively after fine-tuning, causing the classifier unable to distinguish spurious and genuine tokens.

	Spurious score		
Method	film	movie	people
Spuriousness	✗	✓	✓
RoBERTa (Trained on $\mathcal{D}_{biased}$)	0.03	67.4	28.72
RoBERTa (Trained on $\mathcal{D}_{unbiased}$)	0.03	0.09	2.79

Table 3: Neighborhood statistics of target tokens. Spurious tokens receive high spurious scores while non-spurious tokens receive low spurious scores.

shows that fine-tuning causes language models to pull the representations of *book* and *movie* apart and align them with the genuine tokens. In other words, the tokens *book* and *movie* lose their meaning during fine-tuning.

To view this phenomenon in a quantitative manner, we define *spurious score* of a token by the mean probability change of class 1 in the prediction of when inputting the top K neighbors[2], $\mathcal{N}_i$, to f^* . i.e.,

$$\frac{1}{K}\sum_{i=1}^{K}|f^*(\mathcal{N}_i^{\theta_0}) - f^*(\mathcal{N}_i^{\theta})|. \quad (1)$$

Intuitively, if the polarities of the nearest neighbors of a token change drastically (hence obtaining a high spurious score), the token might have lose its original semanticity and is likely to be spurious. We consider only the probability change of class 1 because both tasks presented in this work are binary classifications.

Table 3 revealed that the upper bound model that trained on $\mathcal{D}_{\text{unbiased}}$ change the polarity of the neighbors very slightly and therefore the target tokens have a low spurious score. On the contrary, standard fine-tuning terribly increases the spurious score of the target tokens. The spurious score of non-spurious token (*film* in Amazon binary) remains low regardless of the datasets used in fine-tuning. This hints us the fact that keeping a low spurious score is crucial to learning a robust model.

3 Don't Forget your Language

As we identify with neighborhood analysis that the heart of the problem is the misalignment of spurious tokens and genuine tokens in the language model, we propose a family of regularization techniques, NFL to restrict changes in either parameters or outputs of a language model. Our core idea is to protect our model from spurious correlations with off-the-shelf pretrained language models which are not exposed to spurious correlations. The followings are the variations of NFL:

- NFL-F (**F**rozen). A simple baseline method is setting the weights of the language model to be *frozen* and using the language model as a fixed feature extractor.
- NFL-CO (**C**onstrained **O**utputs). A straightforward idea is to minimize the cosine distance between the representation of each token produced by the language model and that of the initial language model. So we have the regularization term

$$\sum_{m=1}^{M}\text{cos-dist}(\mathcal{M}_{\theta}(w_{i,m}), \mathcal{M}_{\theta_0}(w_{i,m})). \quad (2)$$

- NFL-CP (**C**onstrained **P**arameters). Another

[2]We set K to 100 in our analysis.

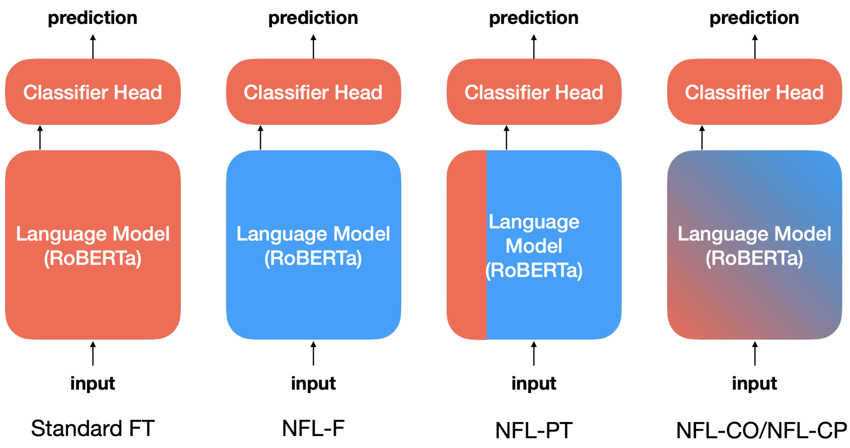


Figure 2: Comparison of fine-tuning and NFL. Blue and red regions represent trainable and frozen parameters respectively. Standard fine-tuning: every parameter is trainable; NFL-F: only the classification head is trainable; NFL-PT: The continuous prompts and the classification head are trainable; NFL-CO/NFL-CP: every parameter is trainable but changes in the language model are restricted by the regularization term in the loss function.

strategy to restrict the language model is to penalize changes in the parameters of the language model. This leads us to the regularization term

$$\sum_i (\theta^i - \theta_0^i)^2. \tag{3}$$

- NFL-PT (**P**rompt-**T**uning). Prompt-tuning introduces trainable continuous prompts while freezing the parameters of the pretrained language model. Therefore, it partially regularizes the output embeddings. In this work, we consider the implementation of Prompt-Tuning v2 (Liu et al., 2022).

	Spurious score		
Method	film	movie	people
Spuriousness	✗	✓	✓
Trained on $\mathcal{D}_{biased}$			
RoBERTa	0.03	67.4	28.72
NFL-CO	0.01	2.28	1.91
NFL-CP	0.01	4.83	2.00
Trained on $\mathcal{D}_{unbiased}$			
RoBERTa	0.03	0.09	2.79

Table 4: Neighborhood statistics of target tokens. NFL achieve low spurious score in spurious tokens.

3.1 Experiment Results

We compare NFL with standard fine-tuning from two aspects: spurious score and robust accuracy. Datasets and models as well as the details of neighborhood statistics are specified in Section 2. The main takeaway is any sensible restriction on the language model to preserve the semanticity of each token is helpful in learning a robust model. Figure 2 summarizes techniques in NFL and compares them with ordinary fine-tuning side-by-side. The weights of the regularization terms in NFL-CO and NFL-CP are discussed in Appendix B.

Spurious Score The effectiveness of NFL is supported by Table 4. Both NFL-CO and NFL-CP achieve a low spurious score for spurious tokens. *book* and *movie* remains in proximity and the polarities of their neighbors alter very slightly after fine-tuning Figure 4. This experiment is not applicable to NFL-F/NFL-PT because they would get a spurious score of 0 by fixing the language model.

Robust Accuracy We call the test accuracy on $\mathcal{D}_{\text{biased}}$ biased accuracy. The robustness of the model is evaluated by the challenging subset $\hat{\mathcal{D}}_{\text{unbiased}} \subset \mathcal{D}_{\text{unbiased}}$ where every example contains at least one of the spurious tokens. The accuracy on this subset is called robust accuracy. The gap between biased accuracy and robust accuracy tells us how much degradation the model is suffering. Table 5 show that while standard fine-tuning is suffering a random-guessing accuracy, NFL enjoys a low degradation and high robust accuracy. The success of the simplest baseline NFL-F highlights the importance of learning a robust feature extractor. While the in-distribution predictive capability of NFL-F is limited by the lack of trainable parameters, other variants of NFL achieve a balance between limiting the model and learning useful features. The best-performing NFL even achieves a robust accuracy that is close to the upper bound.

	Amazon binary			Jigsaw		
Method	Biased Acc	Robust Acc	Δ	Biased Acc	Robust Acc	Δ
Trained on $\mathcal{D}_{biased}$						
RoBERTa	95.7	53.3	-42.4	86.5	50.3	-36.2
NFL-F	89.5	77.3	-12.2	75.3	70.3	-5.0
NFL-CO	92.9	85.7	-7.2	78.9	73.4	-5.5
NFL-CP	95.3	91.3	-4.0	84.8	80.9	-3.9
NFL-PT	94.2	92.9	-1.3	82.5	78.2	-4.3
Trained on $\mathcal{D}_{unbiased}$						
RoBERTa	94.8	95.6	0.8	85.2	82.2	-3.0

Table 5: Results of Amazon binary and Jigsaw. The robustness gap, Δ is given by Robust Acc $-$ Biased Acc. NFL enjoys a low degradation when being exposed to spurious correlations.

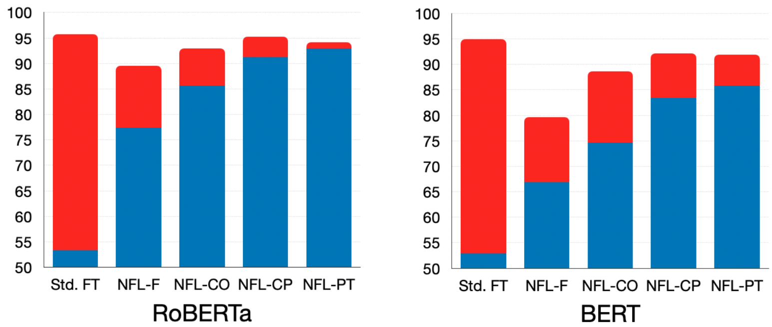


Figure 3: Results of Amazon binary with different pretrained language models. Blue bars represent robust accuracies and red bars represent robustness gaps. The robustness gaps in RoBERTa is smaller than that of BERT.

3.2 Comparison Between Pre-trained Language Models

It is known that RoBERTa is more robust than BERT due to the larger and diversified pretraining data (Tu et al., 2020). As NFL is essentially using the off-the-shelf pretrained language model to protect the main model, we test a hypothesis that language models with richer pretraining are more capable of protecting the main model. Our claim is supported by the experiments shown in Figure 3. While NFL is useful across different choices of pretrained language models, the robustness gap is smaller in RoBERTa than that of BERT when using a regularization term.

4 Naturally Occuring Spurious Correlations

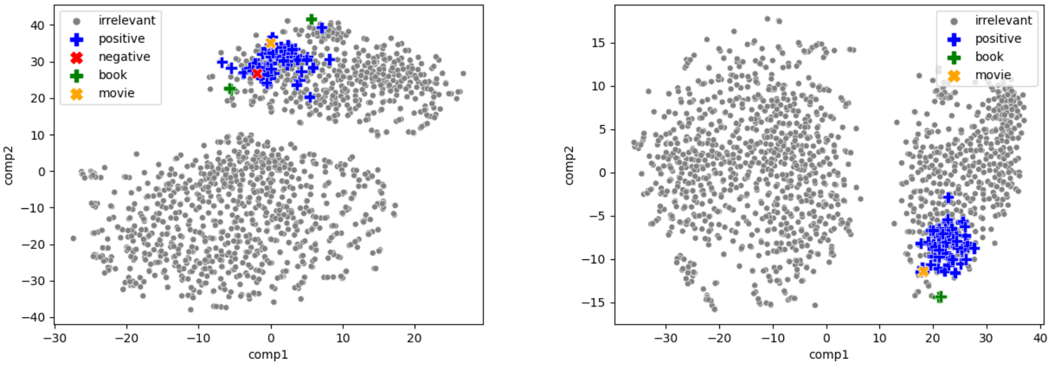
We continue to study naturally occurring spurious correlations with our neighborhood analysis. Spurious correlations are naturally present in datasets due to various reasons such as annotation artifacts, flaws in data collection and distribution shifts (Gururangan et al., 2018; Herlihy and Rudinger, 2021; Zhou et al., 2021). Previous work (Wang and Culotta, 2020; Wang et al., 2022) pointed out in SST2, the token *spielberg* has high co-occurrences with positive but the token itself does not cause the label to be positive. Therefore it is likely to be spurious. Borkan et al. (2019) revealed that models tend to capture the spurious correlations in the toxicity detection dataset by relating the names of frequently targeted identity groups such as *gay* and *black* with toxic content.

4.1 Dataset

SST2 This dataset consists of texts from movie reviews (Socher et al., 2013). It contains 67,300 training examples. We use 10% of the training data for validations and use the original 872 validation data for testing. **Amazon binary, Jigsaw** We follow the settings introduced in Section 2.2.

4.2 Neighborhood Analysis of Naturally Occuring Spurious Correlations

As shown in Table 6, our framework can explain the spurious tokens pointed out by previous work. These naturally occurring spurious tokens demonstrate similar behavior as that of synthetic spurious tokens, *spielberg* is aligned with genuine tokens of positive movie reviews and the names of targeted identity groups (*gay* and *black*) are aligned with



(a) NFL-CO

(b) NFL-CP

Figure 4: Representations after fine-tuning with NFL-CO/NFL-CP. By preventing the formation of erroneous clusters, NFL can learn robust representations.

Target token	Neighbors before fine-tuning	Neighbors after fine-tuning
spielberg (SST2)	spiel, spiegel, rosenberg, goldberg zimmerman, iceberg, bewild, Friedrich	exquisite, dedicated, rising, freedom important, lasting, leadings, remarkable
gay (Jigsaw)	beard, bomb, dog, wood, industrial moral, fat, fruit, cam, boy	whites, lesbians, fucked, black foreigner, shoot, arse, upsetting, die
black (Jigsaw)	white, racist, brown, silver, gray green, blue, south, liberal, generic	ass, demon, fuck, muslim, intellectual populous, homosexual, fools, obnoxious
Canada (Jigsaw)	Spain, Australia, California, Italy Britain, Germany, France, Brazil, Turkey	hypocrisy, ridiculous, bullshit, fuck, stupiddamn, morals, idiots, pissed

Table 6: Nearest neighbors of the spurious tokens before and after fine-tuning. Words in red are associated with negative/toxic labels while words in blue are associated with positive labels according to human annotators.

offensive words as well as other targeted names.

4.3 Detecting Spurious Tokens

There has been a growing interest in detecting spurious correlations automatically to enhance the interpretability of models' prediction. Practitioners may also decide whether they need to collect more data from other sources or simply masking the spurious tokens based on the results of detection. (Wang and Culotta, 2020; Wang et al., 2022; Friedman et al., 2022). In this section, we show that our proposed spurious score can also be used to detect naturally occuring spurious tokens. As we do not have access to a f^* that is trained on $\mathcal{D}_{\text{unbiased}}$ in this setting, we simply use the model fine-tuned on the potentially biased dataset that we would like to perform detections. We compute the spurious score of every token according to Equation 1. The tokens with largest spurious score are listed in Table 7, where the genuine tokens are filtered by human annotators. Take the top spurious token *Canada* as an example, our observation of the changes in neighborhood analysis still holds true (Table 6). The precision of our detection scheme for top 10/20/30 spurious tokens are evaluated by human annotators and listed in Table 8.

5 Related Work

5.1 Mitigating Spurious Correlations

Existing mitigation approaches can be classified into two categories—data-based and model-based (Ludan et al., 2023). Data-based approaches modify the datasets to eliminate spurious correlations. (Goyal et al., 2017; Sharma et al., 2018; McCoy et al., 2019; Zellers et al., 2019) Model-based approaches aim to make the models less vulnerable to spurious correlations by model ensembling and regularization (He et al., 2019; Sagawa et al., 2020; Utama et al., 2020a; Zhao et al., 2022). These prior work under the assumption that the spurious correlations are known beforehand but it is arduous to obtain such information in real-world datasets.

To make the setting more realistic, some existing work do not assume having the information of spurious correlations during training but they do

Top naturally occuring spurious tokens in each dataset	
SST2	allow, void, default, sleeps, not, problem, taste, bottom
Amazon	liberal, flashy, reck, reverted, passive, average, washed, empty
Jigsaw	Canada, witches, sprites, rites, pitches, monkeys, defeating, animals

Table 7: List of top spurious tokens according to their spurious scores verified by human annotators.

	Precision		
Method	Top 10	Top 20	Top 50
Ours			
SST2	0.60	0.50	0.53
Jigsaw	0.50	0.45	0.43
Amazon	0.50	0.40	0.40
Wang et al. (2022)			
SST2	0.40	0.35	0.32

Table 8: Precision of the top detected spurious tokens according to human annotators.

rely on a small set of unbiased data where spurious correlations do not hold for validations and hyper-parameter tuning (Liu et al., 2021; Clark et al., 2020; Utama et al., 2020b). They also further make assumptions on the properties of spurious correlations and prevent models from learning such patterns. Clark et al. (2020) leverage a shallow model to capture overly simplistic patterns. However, Zhao et al. (2022) find that there is not a fixed capacity shallow model that can capture the spurious correlations and determining an appropriate shallow model is also difficult without the information of spurious correlations. NFL takes a new route and tackles the problem by preserving the semantic knowledge in language models, without relying on the simplicity bias assumption.

In a recent study, Kirichenko et al. (2023) claim that the features learned by standard empirical risk minimization (ERM) is good enough and models' performance can be recovered just by re-training the classification layer on the small set of unbiased data. On the contrary, NFL is designed to be not requiring any unbiased data, as having such information regardless of using it during training or not, is a huge assumption. Different from the findings in Kirichenko et al. (2023), we discover that spurious correlations in text classification tasks corrupt the feature extractor by aligning the representations of spurious tokens and genuine tokens. Thus, simply reweighting the features learned by ERM is undesired. The comparison between NFL and DFR is presented in Appendix C. NFL can achieve better performance even with less data and without the information of spurious correlations.

5.2 Model-based Detection of Spurious Tokens

In the context of text classification, some of the previous studies aim to detect spurious tokens for better interpretability. They generally work by finding tokens that contribute the most to models' prediction (Wang and Culotta, 2020; Wang et al., 2022), but do not uncover the internal mechanism of how those spurious tokens acquire excessive importance weights and thereby dominate models' predictions. Our neighborhood analysis reveal that spurious tokens acquire excessive importance due to the erroneous alignment with genuine tokens in the embedding space.

In addition, Wang and Culotta (2020) requires human annotated examples of genuine/spurious tokens while Wang et al. (2022) requires multiple datasets from different domains for the same task. As such external data might be too expensive to collect, our work is motivated to use the widely available pretrained language models as an anchor. The comparison with Wang et al. (2022) is presented in Table 8. Our method can detect spurious tokens with similar precision without requiring multiple datasets and hence is a more practical solution.

6 Conclusion

In this paper, we present our neighborhood analysis to explain how models interact with spurious correlation. Through the analysis, we learn that the corrupted language models capture spurious correlations in text classification tasks by mis-aligning the representation of spurious tokens and genuine tokens. The analysis not only provides a deeper understanding of the spurious correlation issue but can additionally be used to detect spurious tokens. In addition, our observation from the analysis allows designing an effective family of regularization methods that prevent the models from capturing spurious correlations by preventing mis-alignments and preserving the semantic knowledge with the help of off-the-shelf pretrained language models.

Limitations

Our proposed NFL family is built on the assumption that off-the-shelf pretrained language models are unlikely to be affected by spurious correlation as the self-supervised learning procedures behind the models do not involve any labels from downstream tasks. Erroneous alignments formed by biases in the pretraining corpora are then beyond the scope of this work. As per our observation in Section 3.2, we echo the importance of pretraining language models with richer contexts and diverse sources to prevent biases in off-the-shelf pretrained language models in future studies.

References

Parikshit Bansal and Amit Sharma. 2023. Controlling learned effects to reduce spurious correlations in text classifiers.

Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2019. Nuanced metrics for measuring unintended bias with real data for text classification.

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.

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 2020. Learning to model and ignore dataset bias with mixed capacity ensembles. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3031–3045, Online. Association for Computational Linguistics.

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

Dan Friedman, Alexander Wettig, and Danqi Chen. 2022. Finding dataset shortcuts with grammar induction. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4345–4363, Abu Dhabi, United Arab Emirates. 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.

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

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the v in vqa matter: Elevating the role of image understanding in visual question answering.

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

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

Christine Herlihy and Rachel Rudinger. 2021. MedNLI is not immune: Natural language inference artifacts in the clinical domain. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 1020–1027, Online. Association for Computational Linguistics.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4411–4421. PMLR.

Nitish Joshi, Xiang Pan, and He He. 2022. Are all spurious features in natural language alike? an analysis through a causal lens. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9804–9817, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. 2023. Last layer re-training is sufficient for robustness to spurious correlations. In *The Eleventh International Conference on Learning Representations*.

Evan Z Liu, Behzad Haghgoo, Annie S Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. 2021. Just train twice: Improving group robustness without training group information. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 6781–6792. PMLR.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68, Dublin, Ireland. Association for Computational Linguistics.

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.

Adian Liusie, Vatsal Raina, Vyas Raina, and Mark Gales. 2022. Analyzing biases to spurious correlations in text classification tasks. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 78–84, Online only. Association for Computational Linguistics.

Josh Magnus Ludan, Yixuan Meng, Tai Nguyen, Saurabh Shah, Qing Lyu, Marianna Apidianaki, and Chris Callison-Burch. 2023. Explanation-based finetuning makes models more robust to spurious cues.

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.

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

Rishi Sharma, James Allen, Omid Bakhshandeh, and Nasrin Mostafazadeh. 2018. Tackling the story ending biases in the story cloze test. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 752–757, Melbourne, Australia. Association for Computational Linguistics.

Chenglei Si, Dan Friedman, Nitish Joshi, Shi Feng, Danqi Chen, and He He. 2023. What spurious features can pretrained language models combat?

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

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

Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. 2020a. Mind the trade-off: Debiasing NLU models without degrading the in-distribution performance. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8717–8729, Online. Association for Computational Linguistics.

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

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Tianlu Wang, Rohit Sridhar, Diyi Yang, and Xuezhi Wang. 2022. Identifying and mitigating spurious correlations for improving robustness in NLP models. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1719–1729, Seattle, United States. Association for Computational Linguistics.

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

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

Xiang Zhang and Yann LeCun. 2017. Which encoding is the best for text classification in chinese, english, japanese and korean? *CoRR*, abs/1708.02657.

Jieyu Zhao, Xuezhi Wang, Yao Qin, Jilin Chen, and Kai-Wei Chang. 2022. Investigating ensemble methods for model robustness improvement of text classifiers. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1634–1640, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yanan Zheng, Jing Zhou, Yujie Qian, Ming Ding, Chonghua Liao, Li Jian, Ruslan Salakhutdinov, Jie Tang, Sebastian Ruder, and Zhilin Yang. 2022. FewNLU: Benchmarking state-of-the-art methods for few-shot natural language understanding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 501–516, Dublin, Ireland. Association for Computational Linguistics.

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

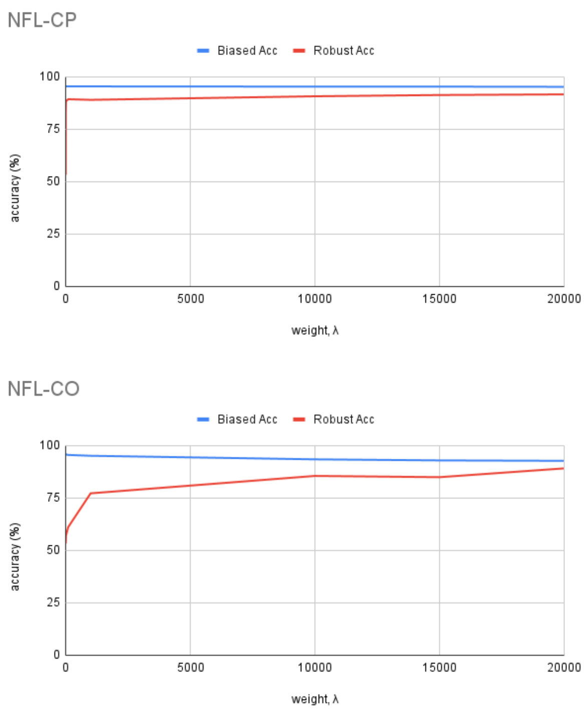


Figure 5: Accuracies of NFL-CP and NFL-CO under different choices of λ.

A Training Details

We use pretrained BERT, RoBERTa and the default hyperparameters in Trainer, offered by Huggingface in all of our experiments. We also use the implementation from Liu et al. (2022) for NFL-PT. The models are trained for 6 epochs except for NFL-PT which takes 100 epochs. The sequence length of continuous prompts in NFL-PT is set to 40. All accuracy reported is the mean accuracy of 3 trials over the seeds {0, 24, 1000000007}.

B Weights of Regularization Terms

In the experiment of Amazon binary, we search the hyperparameter of the weights of NFL-CO and NFL-CP regularization terms over {1, 10, 100, 1000, 10000, 15000, 20000}. Generally there is a trade-off between in-distribution (biased) accuracy and out-of-distribution (robust) accuracy. Nonetheless, we can observe from Figure 5 that as we increase the weights of the regularization term, the drop in-distribution accuracy is insignificant while the improvement in robustness is tremendous. In all of the experiments, we set the weights to be 15000.

Method	Biased Acc	Robust Acc	Δ
Amazon binary			
NFL-PT	94.2	92.9	-1.3
DFR (100%)	93.4	88.9	-4.5
DFR (5%)	93.6	83.1	-9.5
Jigsaw			
NFL-CP	84.8	80.9	-3.9
DFR (100%)	85.9	78.0	-7.9
DFR (5%)	86.3	75.0	-11.3

Table 9: Comparison between NFL and DFR. To avoid repetition with Table 5, we list only the variant of NFL with highest robust accuracy.

C Comparison with DFR (Kirichenko et al., 2023)

Our work consider a setting that our models do not have access to both the information of spurious correlations as well as unbiased data in both training and validation stage. DFR, on the other hand, requires a small unbiased validation set to re-train the classification layer. To reproduce DFR, we use 100%/5% of $\mathcal{D}_{\text{unbiased}}$ to re-train the classifier. Note that DFR would then have access to both $\mathcal{D}_{\text{biased}}$ (during the training of feature extractors) and $\mathcal{D}_{\text{unbiased}}$ (during the re-training of classifiers). As shown in Table 9, NFL indeed achieve a better robust accuracy by robustifying the feature extractor even with less data compared with DFR.