

# Less Learn Shortcut: Analyzing and Mitigating Learning of Spurious Feature-Label Correlation

Yanrui Du<sup>1\*</sup>, Jing Yan<sup>2</sup>, Yan Chen<sup>2</sup>, Jing Liu<sup>2</sup>, Sendong Zhao<sup>1†</sup>,  
Qiaoqiao She<sup>2</sup>, Hua Wu<sup>2</sup>, Haifeng Wang<sup>2</sup>, Bing Qin<sup>1</sup>

<sup>1</sup>Harbin Institute of Technology, Harbin, China

<sup>2</sup>Baidu Inc., Beijing, China

{ yrdu, sdzhao, bqin }@ir.hit.edu.cn

{ yanjing09, chenyan22, liujing46, sheqiaoqiao, wu\_hua, wanghaifeng }@baidu.com

## Abstract

Recent research has revealed that deep neural networks often take dataset biases as a shortcut to make decisions rather than understand tasks, leading to failures in real-world applications. In this study, we focus on the spurious correlation between word features and labels that models learn from the biased data distribution of training data. In particular, we define the word highly co-occurring with a specific label as biased word, and the example containing biased word as biased example. Our analysis shows that biased examples are easier for models to learn, while at the time of prediction, biased words make a significantly higher contribution to the models’ predictions, and models tend to assign predicted labels over-relying on the spurious correlation between words and labels. To mitigate models’ over-reliance on the shortcut (i.e. spurious correlation), we propose a training strategy Less-Learn-Shortcut (LLS): our strategy quantifies the biased degree of the biased examples and down-weights them accordingly. Experimental results on Question Matching, Natural Language Inference and Sentiment Analysis tasks show that LLS is a task-agnostic strategy and can improve the model performance on adversarial data while maintaining good performance on in-domain data.

## 1 Introduction

Pre-trained language models, e.g. BERT [Devlin *et al.*, 2018], ERNIE [Sun *et al.*, 2019] and RoBERTa [Liu *et al.*, 2019b], have achieved great success on many NLP tasks. However, recent studies highlighted that pre-trained models tend to take dataset biases as a shortcut, rather than truly understand tasks [Schuster *et al.*, 2019; Niven and Kao, 2019]. Models’ over-reliance on the shortcut results in their poor generalization ability and low robustness [Geirhos *et al.*, 2020].

The phenomenon of shortcut learning has been widely studied in various NLP tasks. Many previous studies examine

\* The work was done when Yanrui Du was doing internship at Baidu.

† Corresponding author

this phenomenon by constructing artificial adversarial examples, and employ adversarial data augmentation to enhance model robustness [Jia and Liang, 2017; Alzantot *et al.*, 2018; Ren *et al.*, 2019; Jin *et al.*, 2020]. These studies reported high success rates on artificial adversarial examples, but it is uncertain if the models will perform well on real-world data distributions [Morris *et al.*, 2020; Bender and Koller, 2020]. Additionally, recent work [Balkir *et al.*, 2022] indicated that few studies have applied explainable methods to understand or investigate the impact of shortcut learning.

Previous works point out that shortcuts can be traced back to dataset biases [Lai *et al.*, 2021; Gururangan *et al.*, 2018; Kavumba *et al.*, 2021; Du *et al.*, 2021; McCoy *et al.*, 2019; Liu *et al.*, 2019a; Kavumba *et al.*, 2021; Goyal *et al.*, 2017; Ye and Kovashka, 2021; Dawkins, 2021]. For example, if “not” happens to be *contradiction* for most of the training data in Natural Language Inference (NLI) tasks, detecting “not” becomes a successful strategy for models’ prediction, thus leading to an unexpected performance on a shift distribution [Gururangan *et al.*, 2018]. However, most studies are limited to analyzing task-specific shortcuts, which are prohibitive to be transferred to other tasks.

In this work, we analyze the correlations between simple features (e.g. words) and labels, which can be originated from the biased data distribution of any NLP task, to quantitatively investigate the shortcut learning behavior of NLP models. Existing work has argued that, for any NLP task, no single feature on its own should contain information about the labels, and any correlation between simple features and labels is spurious [Gardner *et al.*, 2021]. Based on the above analysis, we propose a task-agnostic training strategy *Less-Learn-Shortcut (LLS)*, which mitigates the shortcut behavior of models, thereby improving their performance on adversarial data.

To examine the spurious feature-label correlation, we first introduce two definitions: biased word, which is the word highly co-occurring with a specific label in a dataset, and biased example, which is the example containing at least one biased word. Then we quantitatively analyze the spurious feature-label correlations on the Question Matching (QM) task. Based on our analysis, we propose our training strategy *LLS*, with which biased training examples are down-weighted according to their biased degrees, and the mod-

els’ over-reliance on the biased words is penalized during fine-tuning. We conduct extensive experiments on QM, NLI and Sentiment Analysis (SA) tasks to evaluate our training strategy and compare it to other task-agnostic strategies such as *Rew.bias* [Utama *et al.*, 2020; Clark *et al.*, 2019] and *Forg.* [Yaghoobzadeh *et al.*, 2021]. Our experimental results demonstrate that *LLS* can improve the model performance on adversarial data while maintaining good performance on in-domain data, and can be easily transferred to different NLP tasks. Additionally, we explore the scenarios in which the above strategies are applicable.

In general, we have the following major findings and contributions:

- We reveal that biased examples (as defined in Sec. 2.2) are easier to be learned than other examples, and with an explainable method LIME [Ribeiro *et al.*, 2016], we find that biased words make significantly higher contributions to models’ predictions than random words (see Sec. 3.1).
- We find that biased words will affect models’ predictions, and that models tend to assign labels highly correlated to the biased words (see Sec. 3.2).
- To mitigate the models’ over-reliance on the spurious correlation, we propose a training strategy *Less-Learn-Shortcut (LLS)*. Experimental results show that *LLS* can improve the models’ performance on adversarial data while maintaining good performance on in-domain data. Furthermore, we compare *LLS* to existing strategies and reveal their respective applicable scenarios. (see Sec. 4).

## 2 Preliminary

In this section, we first introduce the QM datasets on which we analyze the spurious feature-label correlation, then we give the definitions of *biased word* and *biased example*. At last, we provide the settings of our experiments.

### 2.1 Datasets

We conduct our analysis on three datasets, LCQMC, DuQM and OPPO<sup>1</sup>, all of which are about QM task and collected from real-world applications. LCQMC [Liu *et al.*, 2018] is a large-scale Chinese question matching corpus proposed by Harbin Institute of Technology in the general domain BaiduZhidao. DuQM [Zhu *et al.*, 2021] is a fine-grained controlled adversarial dataset aimed to evaluate the robustness of QM models and generated based on the queries collected from Baidu Search Engine<sup>2</sup>. OPPO is collected from OPPO XiaoBu Dialogue application and can be downloaded on CCF Big Data & Computing Intelligence Contest. The data statistics are provided in Tab. 8 (in App. A).

### 2.2 Definitions

Here we provide the definitions we will use in this work. If we denote  $W$  as all words in the dataset, the set of examples containing a specific word  $w_i$  can be formalized as  $S(w_i)$ , and the frequency of  $w_i$  can be formalized as  $f_{w_i}$ . We define

<sup>1</sup>The datasets can be downloaded on <https://luge.ai>.

<sup>2</sup><http://www.baidu.com>.

# Word	# B-word <sub>0</sub>	# B-word <sub>1</sub>	# B-word
58,230	11,145	4,721	15,866

Table 1: The statistics of biased words in LCQMC<sub>train</sub>.

Dataset	# Examples	# B-exp	% B-exp
LCQMC <sub>train</sub>	238,766	98,260	41.15%
LCQMC <sub>test</sub>	12,500	3,246	25.97%
DuQM	10,121	3,264	32.25%
OPPO	10,000	2,498	24.98%

Table 2: The statistics of biased examples in the datasets. B-exp represents biased example.

*biased degree* as  $d_{w_i}^{c_m}$  to measure the degree of word  $w_i$  co-occurring with category  $c_m$  (for QM task,  $c_m \in (0, 1)$ ) and it can be denoted as

$$d_{w_i}^{c_m} = \frac{|S(w_i, c_m)|}{|S(w_i)|} = \frac{|S(w_i, c_m)|}{f_{w_i}} \quad (1)$$

where  $|S(w_i, c_m)|$  represents the number of examples with  $w_i$  and labeled with  $c_m$ .

**Biased word.** A word highly correlated with a specific label in a dataset.<sup>3</sup> To better discuss it, we define *biased word* as the word  $w_i$  with  $f_{w_i} \geq 3$  and  $d_{w_i}^{c_m} \geq 0.8$  for QM task in Sec. 2 and 3. It is worth mentioning that the biased words we analyze in this work are originated from the training set.

We further define *biased word<sub>0</sub>* and *biased word<sub>1</sub>* as the words highly correlated to category 0 and 1. As shown in Tab. 9 (in App. C), “简便” (“handy”) occurs in 35 examples, 33 of which are with category 1, hence it is a biased word<sub>1</sub>. Tab. 1 shows that 27.24% (15864/58230) of words are biased words, and there are more biased word<sub>0</sub> than biased word<sub>1</sub> in LCQMC<sub>train</sub>.

**Biased example.** An example containing at least one biased word. As shown in Tab. 2, 41.15% of examples in LCQMC<sub>train</sub> are biased examples, which are 25.97%, 32.25% and 24.98% in LCQMC<sub>test</sub>, DuQM and OPPO respectively. Since the biased words occur in almost half of the examples in LCQMC<sub>train</sub>, it is meaningful to study their effects on models. The examples without biased words are defined as *unbiased example*.

### 2.3 Experimental Setup

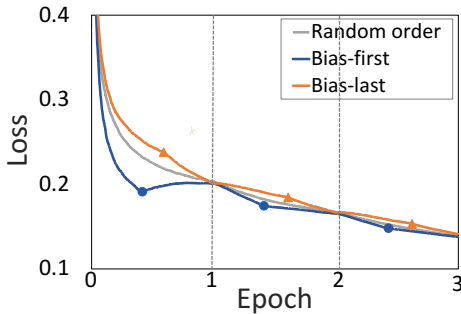
**Models.** We conduct our experiments on three popular publicly available pre-trained models, BERT-base<sup>4</sup>, ERNIE<sub>1.0</sub><sup>5</sup> and RoBERTa-large<sup>6</sup>.

<sup>3</sup>Word is the smallest independent lexical item with its own objective or practical meaning. We use Lexical Analysis of Chinese [Jiao *et al.*, 2018] (<https://github.com/baidu/lac>) for word segmentation in this work.

<sup>4</sup><https://github.com/google-research/bert>.

<sup>5</sup><https://github.com/PaddlePaddle/ERNIE>.

<sup>6</sup><https://github.com/yuncui/Chinese-BERT-wwm>.



(a) Training loss curve of RoBERTa.

Figure 1: Training loss curves of RoBERTa on  $LCQMC_{train}$ , in which  $\bullet$  represents the time of finishing learning biased examples, and  $\blacktriangle$  represents the time of finishing learning unbiased examples.

**Metrics.** As most of the classification tasks, we use accuracy to evaluate the performance of models.

**Training details.** We use the integrated interface BertForSequenceClassification<sup>7</sup> from huggingface for our experiment and use different learning rates for different pre-trained models. Specifically, for  $RoBERTa_{large}$ , the learning rate is 5e-6. For  $BERT_{base}$  and  $ERNIE_{1.0}$ , the learning rate is 2e-5. The proportion of weight decay is 0.01 and the batch size is 64. We train two epochs for  $BERT_{base}$  and  $ERNIE_{1.0}$ , and train three epochs for  $RoBERTa_{large}$ . Every 500 steps, we check the performance of models on  $LCQMC_{dev}$  and choose the checkpoint with the highest accuracy as our main model, and report average results with three different seeds on  $LCQMC_{test}$ , DuQM and OPPO.

### 3 Effect of Feature-Label Correlation

The dataset statistics in Sec. 2 show that 41.15% of examples in  $LCQMC_{train}$  contain biased words. It is a reasonable assumption that the spurious feature-label correlations would affect the models’ behavior and performance. To validate our assumption: 1) we conduct a behavior analysis of the model’s learning and deciding (See Sec. 3.1); 2) we discuss how the feature-label correlation affects the models’ performance by probing the relationship between the biased word and the predicted label (See Sec. 3.2). In Sec. 3.3 we discuss another type of shortcut *word-overlap* and argue that different shortcuts may interact together.

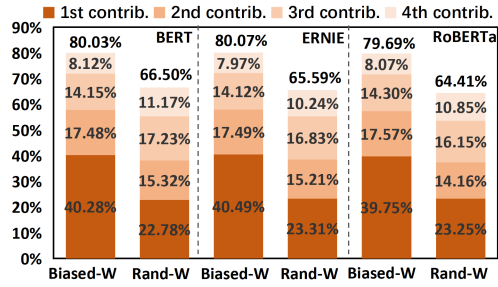
#### 3.1 Feature-Label Correlation and Models’ Behavior

**Models’ learning.** To observe the models’ behavior during training, we separate  $LCQMC_{train}$  into two subsets, biased examples and unbiased examples, and reorganize the train examples in 3 orders:

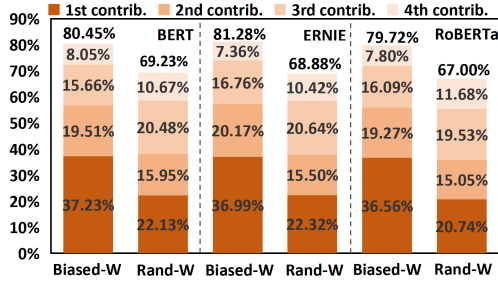
- *bias-first*: firstly biased examples, then unbiased examples;
- *bias-last*: firstly unbiased examples, then biased examples;
- *random order*: shuffle the examples randomly.

We finetune three models (BERT, ERNIE and RoBERTa) in above three orders and plot the training loss curves in Fig. 1

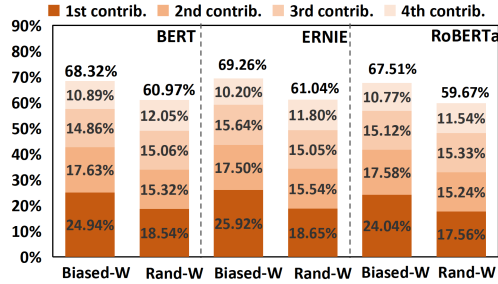
<sup>7</sup><https://huggingface.co/docs/transformers/>.



(a) Results on  $LCQMC_{test}$ .



(b) Results on DuQM.



(c) Results on OPPO.

Figure 2: Probability of biased words and random words with the 1st, 2nd, 3rd, 4th contribution on three test sets. Bias-W and Rand-W represent biased words and random words respectively.

and Fig.4 (see in App B). The training loss curves of all three models present the same tendencies:

- If bias-first, for each epoch, the loss curve drops more rapidly than random order. After learning all the biased examples, the loss curve rises slightly and then decreases.
- If bias-last, the tendency is contrary: for each epoch, the loss drops more slowly than random order until all the unbiased examples have been learned, and then the curve decreases faster.

The above observations reflect that models behave differently when they learn biased examples and unbiased examples: the loss curves of biased examples drop more sharply than other examples, which indicates that the words highly correlated with specific labels are relatively easier for models to learn, and the correlations between words and labels are captured by models as shortcuts.

**Models’ deciding.** In this part, we provide a quantitative analysis of the spurious feature-label correlation’s impact on models’ deciding. If it is easier for a model to learn, will

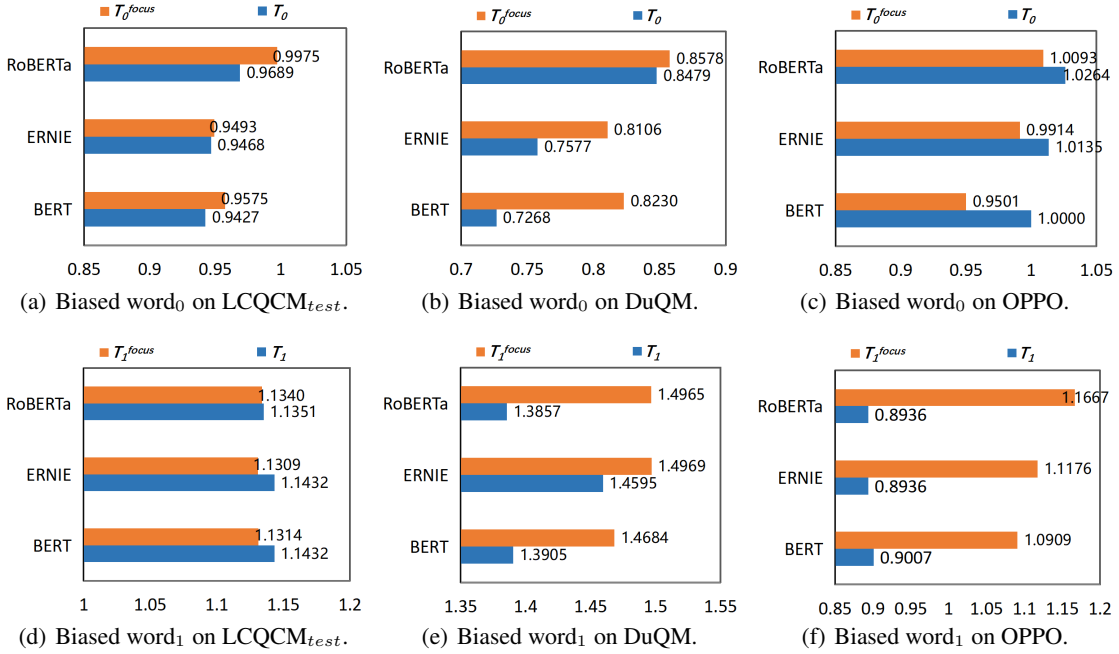


Figure 3: Tendency to predict 0 of biased word<sub>0</sub> and predict 1 of biased word<sub>1</sub>.

the biased words make greater contributions when predicting? Here we select LIME method to measure the contributions of different words in one input to the final prediction, which can interpret the models’ prediction based on locally approximating the model around a given prediction.

To observe the contributions of the biased words, we rank the words based on their contribution scores computing with LIME method. In Fig. 2, we illustrate the ratios of biased words with the highest, second, third, and fourth contribution in three test sets. For comparison, we randomly select words from the input excluding stop-words as random word baseline, and also plot their ratios in Fig. 2. Compared to the random words, the biased words have significantly higher ratios to be ranked among the highest 4, which is about 80% in LCQMC<sub>test</sub> and DuQM, 68% in OPPO.

In summary, the biased examples are easier for models to learn, and the biased words make significantly higher contributions than random words, which implies that models tend to pay more attention to biased words when deciding. With the analysis in this section, we can conclude that the biased word is a shortcut for the models and will affect the models’ behavior. It is therefore substantial to further analyze how it affects the models.

### 3.2 Feature-Label Correlation and Models’ Prediction

Existing works focus only on verifying the existence of shortcut [Bolukbasi *et al.*, 2016; May *et al.*, 2019; Ravfogel *et al.*, 2020; Webster *et al.*, 2020; Kaneko and Bollegala, 2021]. However, there are few quantitative analyses to discuss how the shortcut affects the models’ predictions exactly. In this part, we will focus on probing the relationship between the

biased word and predicted label to analyze how the spurious feature-label correlations impact models. As the biased words are highly correlated to a specific label, it is a reasonable guess that the models tend to assign predicted labels highly correlated to the biased words.

Although the biased words tend to contribute more (discussed in Sec. 3.1), not all biased words make great contribution during predicting. To better analyze the impact of biased word on predicted label, we focus on the testing examples where biased word contributes the most, in which the effects of biased word would be more significant. For convenience, we define the examples in which the biased word makes the greatest contribution as **focus-biased examples**, and we present the statistics of biased examples and focus-biased examples in Tab. 10 and Tab. 11 (see App. D). To measure the tendency of models’ prediction, we define  $T_{c_m}$  as the tendency of model to predict of category  $c_m$ :

$$T_{c_m} = \frac{|S_{pred}(c_m)|/|D|}{|S_{true}(c_m)|/|D|} = \frac{|S_{pred}(c_m)|}{|S_{true}(c_m)|} \quad c_m \in (0, 1) \quad (2)$$

where  $|D|$  represents the number of observed examples,  $|S_{true}(c_m)|$  and  $|S_{pred}(c_m)|$  represent the number of examples with true label  $c_m$  and predicted label as  $c_m$  respectively. Specially, we observe the tendency of models’ prediction on “normal” biased examples and focus-biased examples, and denote them as  $T_{c_m}$  and  $T_{c_m}^{focus}$ . The results are shown in Fig. 3. Fig. 3(a) to Fig. 3(c) show the influence of biased word<sub>0</sub> on three test sets. On DuQM (Fig.3(b)), it is obvious that  $T_0^{focus}$  is higher than  $T_0$  by averaged 7% with all three models, which implies that when biased word<sub>0</sub> contributes the most, models have a high tendency to predict 0. The same result is shown on LCQMC<sub>test</sub> (Fig. 3(a)). However,

Model	Dist. $\leq 1$		Dist. $\leq 2$		Dist. $\leq 3$		Dist. $\leq 4$		Dist. $\leq 5$	
	$T_0$	$T_0^{focus}$	$T_0$	$T_0^{focus}$	$T_0$	$T_0^{focus}$	$T_0$	$T_0^{focus}$	$T_0$	$T_0^{focus}$
BERT	0.739	0.833	0.765	0.821	0.800	0.847	0.866	0.909	0.910	0.933
ERNIE	0.761	0.857	0.779	0.786	0.841	0.847	0.894	0.905	0.946	0.970
RoBERTa	0.870	0.938	0.875	0.932	0.905	0.935	0.950	0.978	0.984	1.004
$\bar{\Delta}$	0.086		0.040		0.028		0.027		0.022	

Table 3: Tendency to predict 0 with edit distance less than 6.  $\bar{\Delta}$  denotes the mean of  $T_0^{focus} - T_0$  on BERT, ERNIE and RoBERTa.

on OPPO (Fig.3(c)),  $T_0$  is slightly higher (0.02~0.05) than  $T_0^{focus}$ . We suppose that it is affected by the co-influencing of another shortcut and we provide an extensive experiment to discuss it in Sec. 3.3. Fig. 3(d) to Fig. 3(f) show the influence of biased word<sub>1</sub>. As shown in Fig. 3(f), models tend to predict 1 when they concentrate on biased word<sub>1</sub> on OPPO, that  $T_1^{focus}$  is higher than  $T_1$  by averaged 26% with all three models. The comparison results on DuQM(Fig.3(e)) show the same tendency for all three models, that  $T_1^{focus}$  is higher than  $T_1$  by averaged 6%. On LCQMC<sub>test</sub> (Fig.3(d)),  $T_1^{focus}$  is almost close to  $T_1$  with all three models.

Overall, we observe that when models pay more attention to biased words, they tend to assign labels over-relying on the biased words. Moreover, to explore why the tendency to 0 is not obvious on OPPO (Fig. 3(c)), we provide a further discussion about the influence of another shortcut *word-overlap*.

### 3.3 Word-Overlap: Another Shortcut for QM Models

In real-world scenarios, different shortcuts may interact together to affect the final prediction. Word overlap shortcut has been widely discussed in many MRC and NLI works [McCoy et al., 2019; Lai et al., 2021; Kaushik and Lipton, 2018]. For QM task, the models tend to predict 0 if a sentence pair has low word overlap, i.e., there are few common words between them, and vice versa. As the result of OPPO shown in Tab. 3, even if models focus on biased word<sub>0</sub>, the tendency to 0 is not significant. We attribute the phenomenon to the word-overlap shortcut in the QM task. To eliminate the influence of word-overlap, we design an experiment on the examples in which the question pairs with high word-overlap. We use *LevenshteinEdit* distance to measure the overlapping degree. We report the models’ prediction tendency with short edit distance in Tab. 3. The results reflect that models have a higher tendency to predict 0 on focus-biased examples than “normal” biased examples, which implies that models tend to predict 0 if we try to eliminate the word-overlap shortcut. Specifically, compared with “normal” biased examples, the average  $T_0^{focus}$  of three models with edit distance 1 increases by 0.086, which is 0.040, 0.028, 0.027 and 0.022 for edit distance of 2, 3, 4 and 5.

Generally, we can deduce that models tend to assign labels relying on the feature-label correlation trick. By eliminating the influence of word-overlap, the models’ prediction tendency towards 0 becomes significant on OPPO. Besides the spurious correlations we study in this work, NLP models

are also affected by many other shortcuts.

## 4 Less-Learn-Shortcut: A Training Strategy to Mitigate Models’ Over-Reliance on Feature-Label Correlation

In Sec. 3 we observe that the spurious feature-label correlation will affect models’ learning and deciding. To mitigate the models’ shortcut learning behavior, we propose a training strategy *Less-Learn-Shortcut (LLS)*, with which all the biased training examples are penalized according to their biased degrees (in Sec. 4.1) during fine-tuning. Most of the existing strategies to mitigate shortcut learning include data augmentation [Jin et al., 2020; Alzantot et al., 2018] and adversarial training [Stacey et al., 2020], which are task-relevant. Our proposed method *LLS* is task-agnostic and can be easily transferred to different NLP tasks.

### 4.1 Reweight Biased Examples

To mitigate the models’ over-reliance on the feature-label correlations, a straightforward idea is to down-weight the biased examples, so that the models are prevented from over-fitting the spurious correlations. In this section, we will introduce how we reweight the biased examples.

**Quantify the impact of correlation.** In Sec. 2, we have defined *biased degree*  $d_{w_i}^{c_m}$  to measure the correlation between the word  $w_i$  and the label  $c_m$ , which can quantify the impact of the correlation. The maximum biased degree of a word among all categories is denoted as  $b_{w_i}^o$  ( $C$  represents all categories).

$$b_{w_i}^o = \max_{c_m \in C} d_{w_i}^{c_m} \tag{3}$$

Furthermore, some existing works show that the word frequency in the training data also influences the models’ prediction [Gu et al., 2020; Cui et al., 2016; Ott et al., 2018]. Considering both biased degree and word frequency, we formulate the impact of a biased word as

$$b_{w_i} = \max_{c_m \in C} d_{w_i}^{c_m} + \alpha f_{w_i} \tag{4}$$

where  $f_{w_i}$  represents the frequency of words  $w_i$  occurring in the training dataset, and  $\alpha$  is a trade-off factor. Then the impact of a biased example can be formulated as the average impact of all biased words it contains:

$$b_e^o = \frac{1}{n} \sum_{i=1}^n b_{w_i} \tag{5}$$

Task		QM			NLI		SA	
Model	Strategy	LCQMC <sub>test</sub>	DuQM	OPPO	SNLI <sub>test</sub>	HANS <sub>test</sub>	Chn <sub>test</sub>	SENTI <sub>robust</sub>
BERT	<i>Finetune</i>	87.16%	67.99%	81.99%	90.80%	57.71%	<b>95.53%</b>	65.77%
	<i>Rew.bias</i>	87.20%	67.79%	81.80%	90.50%	59.12%	94.89%	67.28%
	<i>Forg.</i>	86.84%	68.20%	81.57%	90.48%	<b>61.51%</b>	94.92%	66.88%
	<i>LLS<sub>d</sub></i>	87.40%	68.52%	81.91%	90.69%	59.10%	95.50%	<b>67.51%</b>
	<i>LLS<sub>d+f</sub></i>	87.27%	69.05%	<b>82.08%</b>	90.57%	59.58%	95.00%	67.44%
	<i>LLS</i>	<b>87.86%</b>	<b>69.20%</b>	81.84%	<b>90.82%</b>	59.33%	-	-
ERNIE	<i>Finetune</i>	87.63%	70.08%	82.56%	91.19%	62.59%	<b>96.08%</b>	63.45%
	<i>Rew.bias</i>	87.09%	<b>71.68%</b>	82.44%	91.28%	62.88%	95.64%	63.80%
	<i>Forg.</i>	87.04%	71.62%	82.50%	90.91%	<b>65.22%</b>	95.11%	63.52%
	<i>LLS<sub>d</sub></i>	87.38%	70.61%	82.30%	91.27%	64.26%	95.83%	<b>64.18%</b>
	<i>LLS<sub>d+f</sub></i>	87.61%	70.88%	<b>82.61%</b>	<b>91.31%</b>	64.55%	96.00%	63.55%
	<i>LLS</i>	<b>88.16%</b>	71.65%	82.52%	91.12%	64.76%	-	-
RoBERTa	<i>Finetune</i>	87.58%	72.86%	82.60%	<b>92.54%</b>	74.31%	95.00%	65.32%
	<i>Rew.bias</i>	87.68%	73.91%	82.70%	92.41%	73.14%	95.22%	65.92%
	<i>Forg.</i>	86.50%	73.76%	82.50%	92.25%	74.65%	94.33%	66.07%
	<i>LLS<sub>d</sub></i>	87.85%	74.14%	82.80%	92.47%	74.52%	<b>95.64%</b>	66.17%
	<i>LLS<sub>d+f</sub></i>	87.84%	73.42%	82.71%	92.44%	73.74%	95.28%	<b>66.46%</b>
	<i>LLS</i>	<b>88.46%</b>	<b>74.18%</b>	<b>82.81%</b>	92.42%	<b>74.88%</b>	-	-

Table 4: Performance (accuracy%) of three models on LCQMC<sub>train</sub>, SNLI<sub>train</sub> and Chnsenticorp<sub>train</sub> respectively. As SA task is a single-sentence classification task which not struggle with word-overlap shortcut, we only try *LLS<sub>d</sub>* and *LLS<sub>d+f</sub>* strategy on it. We select *Finetune*, *Rew.bias* and *Forg.* strategies as our baselines. For each model, bold font represents the best performance.

**Exclude the impact of word-overlap for sentence pair tasks.** In Sec. 3.3, we observe that word-overlap is another shortcut in the sentence pair tasks: models tend to give positive predictions to sentence pairs with high word-overlap and negative to low word-overlap. To exclude the impact of word-overlap, when two shortcuts conflict in one biased example, we only consider the impact of the biased word with minimum biased degree:

$$b_e = \begin{cases} \min b_w & , \text{conflict} \\ \frac{1}{n} \sum_{i=1}^n b_{w_i} & , \text{otherwise} \end{cases} \quad (6)$$

**Calculate the loss weight of the biased examples.** We rescale  $b_e$  of all biased examples with the min-max normalization. The loss weights of the biased examples can be denoted as follows, with which the biased examples with higher  $b_e$  will be assigned smaller loss weights:

$$w_{e_j} = 1 - \beta \frac{b_{e_j} - \min b_e}{\max b_e - \min b_e} \quad (7)$$

where  $e_j$  belongs to biased example and  $\beta$  is used for adjusting the low bound of the normalized interval, and the high bound is fixed to 1. It is worth noting that we only reweight the biased examples, and the loss weights of unbiased examples are 1.

## 4.2 Experimental Results

First, we introduce our comparison baseline and then conduct experiments on QM task, NLI task (sentence pair classification task), and SA task (single sentence classification task). Detailed information about the experimental settings can be

Dataset	# Words	# B-word	% B-word
SNLI <sub>train</sub>	42,567	1,261	2.96%
MNLI <sub>train</sub>	101,705	202	0.22%
Chnsenticorp <sub>train</sub>	35,274	4,956	14.05%

Table 5: The statistics of biased words in SNLI<sub>train</sub>, MNLI<sub>train</sub> and Chnsenticorp<sub>train</sub>. B-word represents biased word.

found in App.E. We present the average results of three different seeds and our performance improvements are statistically significant with a p-value of paired t-test less than 0.05.

**Baseline.** In addition to select *Finetune* as our baseline, we re-implement *Rew.bias* [Utama *et al.*, 2020; Clark *et al.*, 2019] and *Forg.* [Yaghoobzadeh *et al.*, 2021] strategies. The core idea of *Rew.bias* is similar to our *LLS* strategy, which is down-weighting biased examples. *Rew.bias* needs to additionally train a bias-only model to score biased examples. *Forg.* strategy uses examples forgotten by the model during training to do secondary training. The above two strategies are both task-agnostic and can be applied to any NLP task.

**QM task.** As shown in Tab. 4, for BERT and RoBERTa, our *LLS* strategy performs best on both the in-domain LCQMC<sub>test</sub> (87.86% and 88.46%) and adversarial DuQM (69.20% and 74.18%). For ERNIE, our *LLS* strategy performs best on in-domain LCQMC<sub>test</sub> (88.16%) and performs close to best on the adversarial DuQM (71.65%). Furthermore, we observe that although *Rew.bias* and *Forg.* strategies improve the model performance on the adversarial DuQM, they only remain performance on LCQMC<sub>test</sub> and OPPO.

Dataset	# Examples	# B-exp	% B-exp
SNLI <sub>train</sub>	549,367	26,590	4.84%
MNLI <sub>train</sub>	392,702	993	0.25%
Chnsenticorp <sub>train</sub>	9,600	9,151	95.11%

Table 6: The statistics of biased examples in SNLI<sub>train</sub>, MNLI<sub>train</sub> and Chnsenticorp<sub>train</sub>. B-exp represents biased example.

By contrast, our *LLS* strategy can improve the model performance on all three test sets.

To better investigate the contributions of different components of *LLS*, we compare *LLS* with two ablations: *LLS<sub>d</sub>* only employs the biased degree to measure the impact of correlation, and does not consider the impact of word-overlap (Eq. 3, 5, and 7); *LLS<sub>d+f</sub>* considers both biased degree and word frequency to measure the correlation, but also does not consider the impact of word-overlap (Eq. 4, 5, and 7). As shown in Tab. 4, *LLS* generally performs the best, indicating that considering word frequencies and excluding word-overlap has a positive effect.

**NLI task.** NLI task aims to determine the relationship between two sentences, whether a premise sentence entails a hypothesis sentence. It is normally formulated as a multi-class classification problem. In our experiments, we try two NLI datasets as the training sets, SNLI [Bowman *et al.*, 2015] and MNLI [Williams *et al.*, 2017]. Tab. 5 and 6 give the statistics of SNLI<sub>train</sub> and MNLI<sub>train</sub>. Although only 2.96% of words in SNLI<sub>train</sub> are biased words, they occur in 4.84% of examples. Compared to SNLI<sub>train</sub>, MNLI<sub>train</sub> is relatively unbiased and contains only 202 biased words (0.22%) and 993 biased examples (0.25%).

We first conduct our experiment on SNLI<sub>train</sub>. We train models on SNLI<sub>train</sub> and evaluate them on the in-domain SNLI<sub>test</sub> and the adversarial HANS<sub>test</sub>. SNLI is a dataset with three classes: entailment, neutral, and contradiction. HANS is a two-class dataset, entailment and non-entailment. As done in previous work [McCoy *et al.*, 2019], to evaluate models on HANS<sub>test</sub>, we convert neutral or contradiction labels to non-entailment. The experimental results are shown in Tab. 4. For BERT and ERNIE, *Forg.* strategy improves the model performance more significantly on the adversarial HANS<sub>test</sub>. We present the statistics of forgotten examples (see App. F), and observe that for the large-scale SNLI<sub>train</sub>, small models such as BERT and ERNIE are more likely to forget examples. Therefore, secondary training with forgotten examples can better help small models increase their robustness. In contrast, for the large RoBERTa model, *Forg.* strategy yields little and our *LLS* strategy performs better. Furthermore, compared to *Finetune* and *Rew.bias* strategies, for all three models, our *LLS* strategy obtains a more significant benefit on the adversarial HANS<sub>test</sub> while maintaining good performance on the in-domain SNLI<sub>test</sub>.

The results on MNLI<sub>train</sub> are shown in Tab. 7. Due to the fact that the MNLI<sub>train</sub> contains fewer biased examples, the effect of *LLS* is not significant. This suggests that *LLS* strategy is more effective for the training set with biased data distribution, helping models learn the spurious correlation less.

Model	Strategy	MNLI <sub>test</sub>	HANS <sub>test</sub>
BERT	<i>Finetune</i>	84.22	52.01
	<i>LLS<sub>d</sub></i>	84.36	<b>52.40</b>
	<i>LLS<sub>d+f</sub></i>	84.31	51.99
	<i>LLS</i>	<b>84.49</b>	52.24

Table 7: Performance (accuracy%) of BERT on MNLI<sub>train</sub>.

**SA task.** SA task aims to determine whether a sentence has a positive or negative sentiment. In our experiment, we train models on Chnsenticorp<sub>train</sub><sup>8</sup> and evaluate them on the in-domain Chnsenticorp<sub>test</sub> and the adversarial Senti<sub>robust</sub> [Wang *et al.*, 2021]. As shown in Tab. 5 and Tab. 6, 14.05% of words in Chnsenticorp<sub>train</sub> are biased words, and they appear in 95.11% of the examples. Unlike NLI and QM tasks, SA task is a single-sentence classification task that is not affected by the word-overlap shortcut, thus we only report the results of *LLS<sub>d</sub>* and *LLS<sub>d+f</sub>* in Tab. 4. Compared to *Rew.bias* and *Forg.* strategies, our *LLS<sub>d</sub>* and *LLS<sub>d+f</sub>* strategies obtain a more significant benefit on the adversarial Senti<sub>robust</sub> and perform better on the in-domain Chnsenticorp<sub>test</sub>. Additionally, it is worth noting that for the small-scale Chnsenticorp<sub>train</sub>, models will not forget too many samples (see App. F) and *Forg.* strategy yields little.

In summary, our proposed *LLS* strategy can significantly improve the model performance on adversarial data while maintaining good performance on in-domain data. Our experiments show that existing strategies struggle to stably improve performance on in-domain data, making further research necessary. Furthermore, we reveal scenarios in which these strategies are applicable. Compared to the *Rew.bias* strategy, *LLS* strategy demonstrates greater advantages on various tasks. However, *LLS* strategy is not applicable for the relatively unbiased dataset, such as MNLI<sub>train</sub>. On the other hand, *Forg.* strategy shows its own advantages on SNLI<sub>train</sub>. Specifically, when training a small model on a large-scale dataset, *Forg.* strategy is a good option to consider.

## 5 Conclusion

In this paper, we explore models’ shortcut learning behavior of spurious correlations between features and labels, and propose a training strategy *LLS* to mitigate the over-reliance of NLP models on the shortcut. Specifically, we observe that the models are prone to learn spurious correlations, and the biased words make significantly higher contributions to models’ predictions than random words. Moreover, we observe that the models tend to be misled by biased words to assign labels. To mitigate the over-reliance on biases, we propose a training strategy *LLS* to penalize the shortcut learning behavior of models. Experimental results show that *LLS* can improve the model performance on adversarial data while keeping good performance on in-domain data, and it is task-agnostic, which can be easily transferred to other tasks. In future research, we will explore how to better measure and formalize the shortcuts in the training data and generalize them as a class of problems.

<sup>8</sup>[https://github.com/pengming617/bert\\_classification](https://github.com/pengming617/bert_classification)

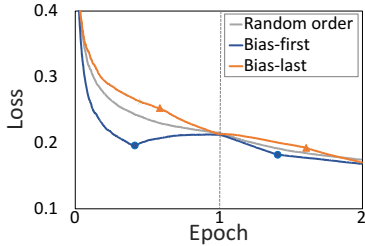
## A Data Statistics

Data statistics are presented in Tab. 8.

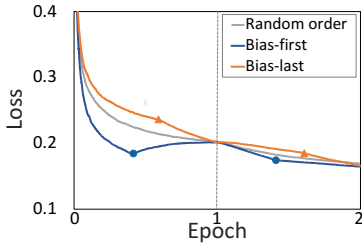
Dataset	Word cnt.			Category		Total
	q1	q2	Total	#0	#1	
$L_{train}$	6.04	6.36	12.40	100,192	138,574	238,766
$L_{test}$	5.51	5.61	11.12	6,250	6,250	12,500
DuQM	4.66	4.80	9.46	7,318	2,803	10,121
OPPO	4.82	4.71	9.53	7,160	2,840	10,000

Table 8: Data statistics.  $L_{train}$  denotes LCQMC training set, and  $L_{test}$  denotes LCQMC test set.

## B Training Loss



(a) Training loss curve of BERT.



(b) Training loss curve of ERNIE.

Figure 4: Training loss curves of BERT and ERNIE on  $LCQMC_{train}$ , in which  $\bullet$  represents finishing learning biased examples, and  $\blacktriangle$  represents finishing learning unbiased examples.

## C Examples of Bias-word

Examples of bias-word<sub>0</sub> and bias-word<sub>1</sub> are given in Tab. 9.

## D Statistics of Bias-Example and Focus-Bias Examples

The statistics of "normal" bias-examples and focus-bias examples are given in Tab. 10 and Tab. 11.

## E Experimental Settings

Experimental settings are introduced Tab. 12.

## F Statistics of Forgotten Examples

In Tab. 13, we record the number of examples forgotten by models on different tasks.

	Word	Category		Total	B-degree
		#0	#1		
<b>B-word<sub>0</sub></b>	漂浮 (float)	5	0	5	1.00
<b>B-word<sub>1</sub></b>	简便 (handy)	2	33	35	0.94

Table 9: Examples of bias-word<sub>0</sub>, and bias-word<sub>1</sub>. B-word<sub>0</sub> represents bias-word<sub>0</sub>, and B-word<sub>1</sub> represent bias-word<sub>1</sub>.

B-word <sub>0</sub>	LCQMC <sub>test</sub>		DuQM		OPPO	
	#S	#S <sub>focus</sub>	#S	#S <sub>focus</sub>	#S	S <sub>focus</sub>
BERT		551		824		474
ERNIE	1,777	554	2,375	879	1,991	494
RoBERTa		517		844		457

Table 10: Statistics of bias-example<sub>0</sub> ( $S$ ) and focus-bias<sub>0</sub> examples ( $S_{focus}$ ). Focus-bias<sub>0</sub> examples represent the examples where bias-word<sub>0</sub> makes great contribution.

B-word <sub>1</sub>	LCQMC <sub>test</sub>		DuQM		OPPO	
	#S	#S <sub>focus</sub>	#S	#S <sub>focus</sub>	#S	S <sub>focus</sub>
BERT		753		336		159
ERNIE	1,543	774	1,095	340	602	163
RoBERTa		722		309		154

Table 11: Statistics of bias-example<sub>1</sub> ( $S$ ) and focus-bias<sub>1</sub> examples ( $S_{focus}$ ). Focus-bias<sub>1</sub> examples represent the examples where bias-word<sub>1</sub> makes great contribution.

Dataset	Model	Epoch	Lr	Bs	Wd
LCQMC	BERT	2	2e-5	64	0.01
	ERNIE	2	2e-5	64	0.01
	RoBERTa	3	5e-6	64	0.01
SNLI	BERT	2	2e-5	64	0.01
	ERNIE	2	2e-5	64	0.01
	RoBERTa	2	5e-6	64	0.01
Chnsenticorp	BERT	5	2e-5	64	0.01
	ERNIE	5	2e-5	64	0.01
	RoBERTa	5	5e-6	64	0.01

Table 12: Introduce our experimental settings. Lr represents learning rate, Bs represents batch size and Wd represents weight decay.

	BERT	ERNIE	RoBERTa
LCQMC	15,589	18,591	17,846
SNLI	55,696	51,397	37,466
Chnsenticorp	1,154	1,161	1,187

Table 13: Statistics of forgotten examples during finetuning. Each value represents how many samples models forget during finetuning. An obvious phenomenon is that on large-scale SNLI dataset, small models tend to forget more samples (55,696 for BERT, 51,397 for ERNIE, but 37,466 for RoBERTa).



## Acknowledgements

This work was supported by the National Key R&D Program of China [2021ZD0113302]; and the National Natural Science Foundation of China [62206079]; and Heilongjiang Provincial Natural Science Foundation of China [YQ2022F006].

## References

- [Alzantot *et al.*, 2018] Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896, Brussels, Belgium, 2018. Association for Computational Linguistics.
- [Balkir *et al.*, 2022] Esma Balkir, Svetlana Kiritchenko, Isar Nejadgholi, and Kathleen C Fraser. Challenges in applying explainability methods to improve the fairness of nlp models. *arXiv preprint arXiv:2206.03945*, 2022.
- [Bender and Koller, 2020] Emily M Bender and Alexander Koller. Climbing towards nlu: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, 2020.
- [Bolukbasi *et al.*, 2016] Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *Advances in neural information processing systems*, 29:4349–4357, 2016.
- [Bowman *et al.*, 2015] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*, 2015.
- [Clark *et al.*, 2019] Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. Don’t take the easy way out: Ensemble based methods for avoiding known dataset biases. *arXiv preprint arXiv:1909.03683*, 2019.
- [Cui *et al.*, 2016] Yiming Cui, Zhipeng Chen, Si Wei, Shijin Wang, Ting Liu, and Guoping Hu. Attention-over-attention neural networks for reading comprehension. *arXiv preprint arXiv:1607.04423*, 2016.
- [Dawkins, 2021] Hillary Dawkins. Marked attribute bias in natural language inference. *arXiv preprint arXiv:2109.14039*, 2021.
- [Devlin *et al.*, 2018] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [Du *et al.*, 2021] Mengnan Du, Varun Manjunatha, R. Jain, Ruchi Deshpande, Franck Dernoncourt, Jiuxiang Gu, Tong Sun, and Xia Hu. Towards interpreting and mitigating shortcut learning behavior of nlu models. In *NAACL*, 2021.
- [Gardner *et al.*, 2021] Matt Gardner, William Merrill, Jesse Dodge, Matthew E Peters, Alexis Ross, Sameer Singh, and Noah Smith. Competency problems: On finding and removing artifacts in language data. *arXiv preprint arXiv:2104.08646*, 2021.
- [Geirhos *et al.*, 2020] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.
- [Goyal *et al.*, 2017] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6904–6913, 2017.
- [Gu *et al.*, 2020] Shuhao Gu, Jinchao Zhang, Fandong Meng, Yang Feng, Wanying Xie, Jie Zhou, and Dong Yu. Token-level adaptive training for neural machine translation. *arXiv preprint arXiv:2010.04380*, 2020.
- [Gururangan *et al.*, 2018] Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R Bowman, and Noah A Smith. Annotation artifacts in natural language inference data. *arXiv preprint arXiv:1803.02324*, 2018.
- [Jia and Liang, 2017] Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. *arXiv preprint arXiv:1707.07328*, 2017.
- [Jiao *et al.*, 2018] Zhenyu Jiao, Shuqi Sun, and Ke Sun. Chinese lexical analysis with deep bi-gru-crf network. *arXiv preprint arXiv:1807.01882*, 2018.
- [Jin *et al.*, 2020] Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is BERT really robust? A strong baseline for natural language attack on text classification and entailment. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8018–8025. AAAI Press, 2020.
- [Kaneko and Bollegala, 2021] Masahiro Kaneko and Danushka Bollegala. Debiasing pre-trained contextualised embeddings. *arXiv preprint arXiv:2101.09523*, 2021.
- [Kaushik and Lipton, 2018] Divyansh Kaushik and Zachary C Lipton. How much reading does reading comprehension require? a critical investigation of popular benchmarks. *arXiv preprint arXiv:1808.04926*, 2018.
- [Kavumba *et al.*, 2021] Pride Kavumba, Benjamin Heinzlerling, Ana Brassard, and Kentaro Inui. Learning to learn to be right for the right reasons. *arXiv preprint arXiv:2104.11514*, 2021.
- [Lai *et al.*, 2021] Yuxuan Lai, Chen Zhang, Yansong Feng, Quzhe Huang, and Dongyan Zhao. Why machine reading comprehension models learn shortcuts? *arXiv preprint arXiv:2106.01024*, 2021.

- [Liu *et al.*, 2018] Xin Liu, Qingcai Chen, Chong Deng, Hua-jun Zeng, Jing Chen, Dongfang Li, and Buzhou Tang. Lcqmc: A large-scale chinese question matching corpus. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1952–1962, 2018.
- [Liu *et al.*, 2019a] Nelson F Liu, Roy Schwartz, and Noah A Smith. Inoculation by fine-tuning: A method for analyzing challenge datasets. *arXiv preprint arXiv:1904.02668*, 2019.
- [Liu *et al.*, 2019b] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.
- [May *et al.*, 2019] Chandler May, Alex Wang, Shikha Bordia, Samuel R Bowman, and Rachel Rudinger. On measuring social biases in sentence encoders. *arXiv preprint arXiv:1903.10561*, 2019.
- [McCoy *et al.*, 2019] R Thomas McCoy, Ellie Pavlick, and Tal Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. *arXiv preprint arXiv:1902.01007*, 2019.
- [Morris *et al.*, 2020] John X Morris, Eli Lifland, Jack Lanchantin, Yangfeng Ji, and Yanjun Qi. Reevaluating adversarial examples in natural language. *arXiv preprint arXiv:2004.14174*, 2020.
- [Niven and Kao, 2019] Timothy Niven and Hung-Yu Kao. Probing neural network comprehension of natural language arguments. *arXiv preprint arXiv:1907.07355*, 2019.
- [Ott *et al.*, 2018] Myle Ott, Michael Auli, David Grangier, and Marc’Aurelio Ranzato. Analyzing uncertainty in neural machine translation. In *International Conference on Machine Learning*, pages 3956–3965. PMLR, 2018.
- [Ravfogel *et al.*, 2020] Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. Null it out: Guarding protected attributes by iterative nullspace projection. *arXiv preprint arXiv:2004.07667*, 2020.
- [Ren *et al.*, 2019] Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1085–1097, Florence, Italy, 2019. Association for Computational Linguistics.
- [Ribeiro *et al.*, 2016] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. ” why should i trust you?” explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.
- [Schuster *et al.*, 2019] Tal Schuster, Darsh Shah, Yun Jie Serene Yeo, Daniel Roberto Filizzola Ortiz, Enrico Santus, and Regina Barzilay. Towards debiasing fact verification models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3419–3425, 2019.
- [Stacey *et al.*, 2020] Joe Stacey, Pasquale Minervini, Haim Dubossarsky, Sebastian Riedel, and Tim Rocktäschel. Avoiding the hypothesis-only bias in natural language inference via ensemble adversarial training. *arXiv preprint arXiv:2004.07790*, 2020.
- [Sun *et al.*, 2019] Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*, 2019.
- [Utama *et al.*, 2020] Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. Towards debiasing nlu models from unknown biases. *arXiv preprint arXiv:2009.12303*, 2020.
- [Wang *et al.*, 2021] Lijie Wang, Hao Liu, Shuyuan Peng, Hongxuan Tang, Xinyan Xiao, Ying Chen, Hua Wu, and Haifeng Wang. Dustrust: A sentiment analysis dataset for trustworthiness evaluation. *arXiv preprint arXiv:2108.13140*, 2021.
- [Webster *et al.*, 2020] Kellie Webster, Xuezhi Wang, Ian Tenney, Alex Beutel, Emily Pitler, Ellie Pavlick, Jilin Chen, Ed Chi, and Slav Petrov. Measuring and reducing gendered correlations in pre-trained models. *arXiv preprint arXiv:2010.06032*, 2020.
- [Williams *et al.*, 2017] Adina Williams, Nikita Nangia, and Samuel R Bowman. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*, 2017.
- [Yaghoobzadeh *et al.*, 2021] Yadollah Yaghoobzadeh, Soroush Mehri, Remi Tachet des Combes, Timothy J Hazen, and Alessandro Sordoni. Increasing robustness to spurious correlations using forgettable examples. In *EACL*, 2021.
- [Ye and Kovashka, 2021] Keren Ye and Adriana Kovashka. A case study of the shortcut effects in visual commonsense reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3181–3189, 2021.
- [Zhu *et al.*, 2021] Hongyu Zhu, Yan Chen, Jing Yan, Jing Liu, Yu Hong, Ying Chen, Hua Wu, and Haifeng Wang. Duqm: A chinese dataset of linguistically perturbed natural questions for evaluating the robustness of question matching models. *arXiv preprint arXiv:2112.08609*, 2021.