An Empirical Study on Robustness of Language Models via Decoupling Representation and Classifier

Anonymous ACL submission

Abstract

Recent studies indicate that shortcut learning behavior exists in language models, and thus a number of mitigation methods are proposed, such as advanced PLMs and debiasing methods. However, few studies have explored how different factors affect the robustness of language models. To bridge this gap, we study the different PLMs and analyze the effect of representations and classifiers on robustness using probing techniques on the NLU tasks. First, we find that the low robustness of language models is not due to the inseparability of representations on the challenging dataset. Second, we find that a potential reason for the difficulty in improving the robustness of language models is the significantly high similarity between the representations with opposite semantics from in-distribution and out-of-distribution. Third, we find that debiasing methods are likely to distort representations and merely improve performance by better classifiers in some cases[1]. Finally, we propose a probing tool to measure the impact on the robustness of language models from representations and classifiers using the decoupled training strategy with debiasing methods. In addition, we conduct extensive experiments on real-world datasets, suggesting the effectiveness of the proposed methods.

1 Introduction

Pre-trained Language Models (PLMs), such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), have achieved state-of-the-art results for Natural Language Understanding (NLU) tasks. Despite their successes, recent studies show the phenomenon that PLMs are prone to learning superficial surface patterns that are spuriously associated with the target label, and to make use of biases/artifacts from the dataset as shortcuts for prediction (Gururangan et al., 2018; McCoy et al., 2019; Utama et al., 2020b), which is defined as shortcut learning (Geirhos et al., 2020). For a common NLU task, Natural Language Inference (NLI), the shortcut learning behavior is defined as that the model achieves high accuracy only by using specific words but not understanding the language (Naik et al., 2018; Sanchez et al., 2018; Du et al., 2021). As a result, the models perform poorly on out-of-distribution (OOD) examples.

The quality of representations is widely considered to be the key reason for the shortcut learning and poor generalization ability of the NLU models. Because of this, there is a large body of literature that analyzes and understands the learned representation (Perone et al., 2018; Krasnowska-Kieraś and Wróblewska, 2019; Pruksachatkun et al., 2020; Mendelson and Belinkov, 2021). Unlike previous work, we intend to answer the following research questions: 1) *Whether the low robustness of language models is primarily due to representations not being easily separable?* 2) *What is the relative role of the representation and the classifier in the low robustness of language models?*

In this work, we apply the t-SNE visualization technique and DirectProbe (Zhou and Srikumar, 2021) probing technique to representations that are from five PLMs: BERT, RoBERTa, BART (Lewis et al., 2020), ELECTRA (Clark et al., 2020b), and DeBERTa (He et al., 2021). Meanwhile, we present a new probing strategy based on debiasing methods. Based on the above techniques and strategies, our findings on the robustness of language models are briefly described below.

To begin with, we visualize the representations of the above five pre-trained language models. We find that not only [CLS] but also [MEAN] embeddings form clearer boundaries between representations of different labels, despite only [CLS] embeddings are fed into the classifiers (§4.1).

Then, we investigate the geometric structure of representations. Via DirectProbe probing tech-

[1] In this work, we denote the **representation** as the output of PLMs and **classifier** as the several fully connected layers that are used to serve the classification purpose.

nique, we obtain clusters with only the same label representations in each cluster. We find that the number of clusters is equal to the number of label categories in most cases, which means that the representations are linearly separable (§4.2).

Furthermore, we further study the properties of representations by computing the similarity between the clusters with opposite semantics on MNLI and HANS. We find that a possible reason why it is not easy to improve the robustness of models is that the representations with opposite semantic labels are too similar (§4.3).

Finally, we investigate the effect of encoders and classifiers on the robustness of language models respectively. Using debiasing methods as a probing tool, we find that both the representation and the classifier of the models play a significant role in the shortcut learning behavior for the NLI task. Furthermore, we find that debiasing methods do not always improve the quality of representations. Instead, they only improve performance by optimizing the classifiers in some cases (§4.4).

2 Preliminaries

In this work, we probe the representations and classifiers of PLMs after fine-tuning. To this end, we briefly introduce two techniques that we use in our analyses: the probing method and the ensemble-based debiasing framework.

2.1 Probing Method

Normally, trained classifiers are used as probes to understand the quality of the information encoded in the representation, which are trained with the encoders frozen (Hewitt et al., 2021; Whitney et al., 2021; Belinkov, 2021). However, the classifier probes focus only on the performance of the target task and cannot clarify the representation in detail (Zhou and Srikumar, 2022). To this end, we apply a probing technique named DirectProbe (Zhou and Srikumar, 2021) instead of classifier probes. It can provide a fine-grained analysis of the representation from a geometric perspective.

The representation in the form of embeddings is fed into DirectProbe, and the number of clusters is returned. These clusters satisfy the condition that the example points contained in a cluster must have the same label and that there are no overlaps between any two clusters (that is, there exists a separator between the two sets of example points). We can learn about various linguistic attributes of the representation by measuring the properties of the corresponding clusters. In this work, we focus on an important property: the number of clusters.

Number of Clusters The number of clusters can quantify the linear separability of the representation for a special task. In particular, when the number of clusters equals the number of label categories, the embeddings of examples with the same label are close enough in the semantic representation space. In this ideal case, the models can achieve perfect performance with a simple linear classifier. In contrast, when the number of clusters is more than the number of label categories, the example points with the same label are grouped into at least two clusters. This suggests a complex geometric structure of representations, and a complex classifier is needed to achieve desirable performance.

2.2 Ensemble-based Debiasing Framework

The ensemble-based debiasing framework (EBD) (Xiong et al., 2021) is generally used to mitigate the shortcut learning behavior of the NLU models. This framework has the advantage that it is a model-agnostic debiasing framework, which makes it possible to debias models adaptively. EBD framework consists of bias-only models and debiasing methods. Bias-only models are used to make the main models perform debiasing training by adjusting the learning target. Debiasing methods provide strategies on how to debias the main models in practice. The EBD framework is commonly formalized as a two-stage method (Clark et al., 2019; Sanh et al., 2021). In the first stage, the bias-only model is trained to recognize simple and hard examples. In the second stage, the main models are trained as an ensemble with the bias-only model according to the selected debiasing methods.

2.2.1 Bias-only Model

Recently, several works in the literature have proposed exploring bias-only models to improve the performance of the EBD framework. For example, Utama et al., 2020b, Sanh et al., 2021, and Clark et al., 2020a try to reduce the need for a prior knowledge on bias or shortcut. In these works, they obtain bias-only models with two different strategies: i) training a copy of the main model with a small random subset of training examples for a few epochs; and ii) using a shallow or small model with limited capacity. In our work, we take the first strategy, and more details are given in Appendix A.

In the following, we describe the workflow of bias-only models. For clarity, we denote the bias-only model by f_b. Given an example (x^i, y^i) in the training dataset, we denote the output of f_b as $f_b\left(x^i\right) = {p_b}^i$. Probability ${p_b}^i$ can quantify how much the model learns about shortcut features from example (x^i, y^i) (i.e., how likely this example contains biases). Specifically, the extent to which models learn shortcut features can be evaluated by ${p_b}^{(i,c)}$ which denotes the probability of ${p_b}^i$ on the label y^i, where c is the index of the correct category in the label y^i. For example, when ${p_b}^{(i,c)}$ is closer to 1 (i.e., the bias-only model is more confident about the example x^i on the label y^i), the model learns more potentially shortcut features. Instead, when ${p_b}^{(i,c)}$ is closer to 0, the bias-only model is more unconfident about the example x^i on the label y^i. As such, the example x^i is likely to be a hard example to which the model is supposed to pay more attention during training.

2.2.2 Debiasing Method

We first denote the main model by f_d parameterized by θ_d, and then use the bias-only model f_b obtained in §2.2.1 to perform debiasing training on f_d. In this work, we mainly investigate two common model-agnostic debiasing methods: sample re-weighting (Schuster et al., 2019) and product-of-experts (Clark et al., 2019; Karimi Mahabadi et al., 2020). In the following, we describe the implementation details of these two methods.

Example Re-weighting Example re-weighting is a simple yet effective debiasing method. It can be briefly summarized as re-weighting the importance of a given training example (x^i, y^i) by directly assigning a weight to the example (x^i, y^i). The weight is formalized as $1 - p_b^{(i,c)}$. Thus, the individual loss of the example (x^i, y^i) for the parameters θ_d is defined as follows:

$$\mathcal{L}\left(\theta_d\right) = -\left(1 - {p_b}^{(i,c)}\right) y^i \cdot \log p_d,$$

where p_d is the $softmax$ output of the main model f_d. Here, we regard training samples with high probability by the bias-only model as biased/shortcut samples. When the bias-only model assigns a high probability to ${p_b}^{(i,c)}$, the contribution of a training example to $\mathcal{L}\left(\theta_d\right)$ is reduced.

Product-of-Experts In this method, the main model (i.e., a debiased model) is trained in an ensemble with a bias-only model. Specifically, the $softmax$ outputs of the main model f_d and the bias-only model f_b are combined to form new predictions. Then they are used to calculate the new loss while optimizing the parameters θ_d. The individual loss of the example (x^i, y^i) for the parameters θ_d is defined as follows:

$$\mathcal{L}\left(\theta_d\right) = -y^i \cdot \log softmax\left(\log p_d + \log p_b\right).$$

During training with debiasing methods, the parameters of the bias-only model f_b are frozen to lower the importance of biased examples in training loss, and only the parameters of the main model f_d are optimized. During the inference time, only the prediction probability of f_d is used.

3 Experimental Setup

3.1 Tasks & Datasets

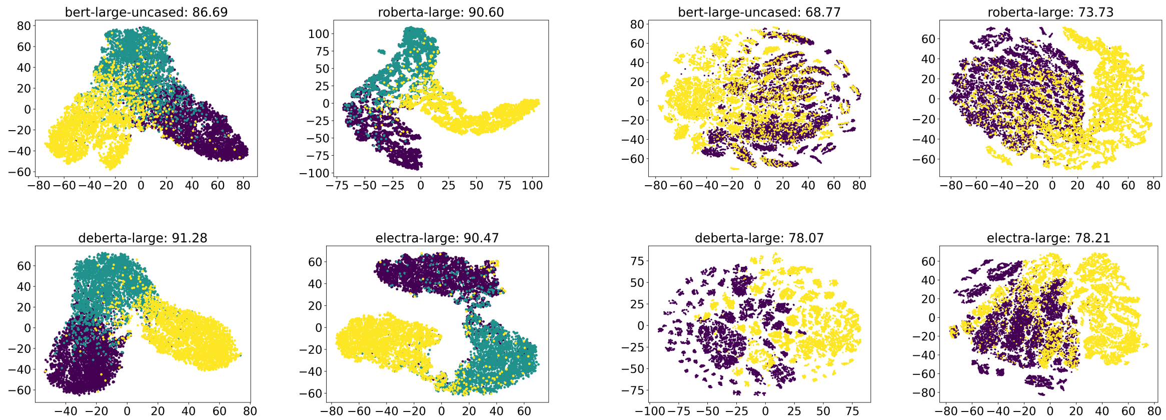
In this work, we focus on a common NLU task: Natural Language Inference (NLI), the model of which is presented with a pair of sentences and asked to return the relationship between their meanings (Williams et al., 2018). A pair of sentences contains a premise sentence and a hypothesis sentence. The relationship between their meanings is one label of *entailment*, *neutral*, and *contradiction*.

MNLI MNLI (Williams et al., 2018) is divided into training dataset, matched development dataset, and mismatched development set. The training dataset and the matched development dataset are derived from the same five genres, and the mismatched development dataset are derived from the other five genres. Typically, we first use the training dataset to train an NLU model, which consists of 392,702 instances. Then, we use the matched development dataset to choose an optimal NLU model, which consists of 9,815 instances.

HANS HANS (McCoy et al., 2019) has been proposed to evaluate whether models have learned statistical patterns or semantic understanding and reasoning. It focuses on three heuristics: the lexical overlap heuristic, the subsequence heuristic, and the constituent heuristic. HANS consists of 30,000 synthetic instances, and distributes 10,000 ones to each of the heuristics. Note that HANS is only used to evaluate the models and not to train the models or adjust the hyperparameters.

3.2 NLU Models

We conduct the empirical study on five different kinds of pre-trained model: BERT, RoBERTa,



(a) [CLS] in MNLI

(b) [CLS] in HANS

Figure 1: The t-SNE visualization result of [CLS] embeddings from different pre-trained language models on MNLI and HANS. The words above figures are the selected model and corresponding accuracy.

BART, ELECTRA, and DeBERTa. These models are used as encoders for the NLU models, which can provide contextual word embeddings. We use the corresponding pre-trained models and fine-tuned models from Huggingface Transformers[2]. The input fed into the encoders of the NLU models is a pair of concatenated premise sentences and hypothesis sentences, which are separated by a special [SEP] token. Then, we obtain sentence pair representations through these models, which are the [CLS] embeddings from encoders. These representations are fed into the classification head (that is, the classifier) of the NLU models. Here, the classification head takes a simple architecture, two linear layers with the activation function.

3.3 Implementation Details

For all pre-trained models, we fine-tune the models without debiasing methods and with the example re-weighting debiasing method for 3 epochs. We find that the models converge slowly when fine-tuning the models with the product-of-experts debiasing method. Thus, we follow He et al. (2019) to fine-tune longer, i.e., 6 epochs. We use AdamW optimizer (Loshchilov and Hutter, 2019) with the default learning rate $5 * 10^{-5}$, where the betas are set as [0.9, 0.999] and the L2 weight decay is set to 0.01. We set the batch size to 32 and warmup ratio to 0.1. All experiments are run with 3 random seeds and the average values are reported, which are completed on the work station with 2 Nvidia 2080Ti GPUs.

[2] https://huggingface.co/models

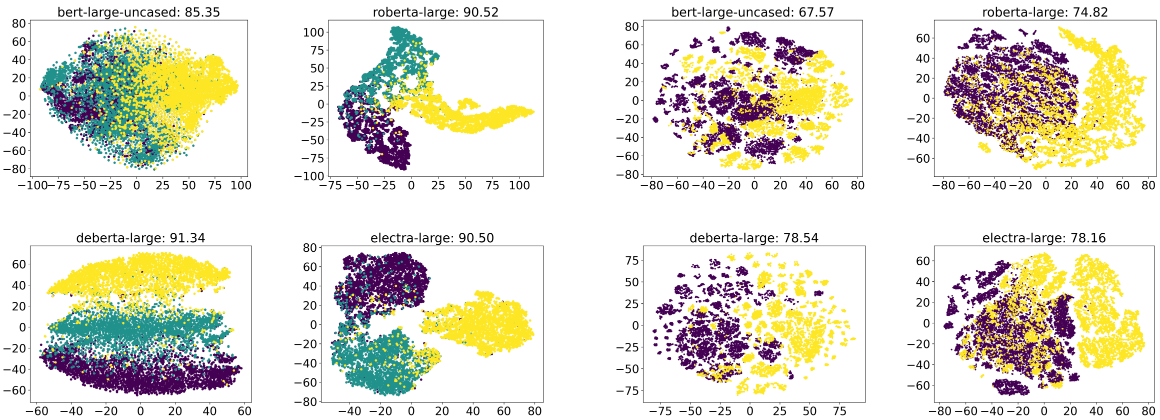
4 Experimental Analysis

In this section, we first use the visualization technique to investigate the separability of representations (§4.1). Then we investigate the linear separability of representations using DIRECTPROBE (§4.2). Furthermore, we propose an explanation for the difficulty in improving robustness by computing the similarity of representations (§4.3). Finally, we analyze the effect of representations and classifiers on robustness by considering debiasing methods as a probing tool (§4.4).

4.1 Visualization of Representations

To investigate the semantic representation space learned by the model, we extract embeddings of the special classification token [CLS] in the final hidden state and visualize them using t-SNE (Van der Maaten and Hinton, 2008). Figures 1(a) and 1(b) show the visualization results for MNLI and HANS, respectively. We show only the results of $\text{BERT}_{\text{large}}$, $\text{RoBERTa}_{\text{large}}$, $\text{DeBERTa}_{\text{large}}$ and $\text{ELECTRA}_{\text{large}}$. Appendix B includes the results for more models. Based on the results, we find that the better performance the model achieves, the clearer the boundaries the representation forms. We believe that this is the reason why high performance is achieved with only a simple two-layer MLP network as the classifier.

In addition, we visualize the mean embeddings of all tokens in the final hidden state, which are abbreviated as [MEAN]. Note that the models are not trained with [MEAN]. The results of MNLI and HANS are shown in Figures 2(a) and 2(b), respec-



(a) [MEAN] in MNLI

(b) [MEAN] in HANS

Figure 2: The t-SNE visualization result of [MEAN] embeddings from different pre-trained language models on MNLI and HANS. The words above figures are the selected model and corresponding accuracy with [MEAN].

tively. Similarly, we show the results for more models in Appendix B. We observe that the [MEAN] embeddings can also form clearer boundaries as the models achieve better performance, despite the [CLS] embeddings are fed into classifiers. Meanwhile, we freeze the encoders of the models trained with [CLS] and feed [MEAN] into the classifiers, the performance of which is close to the original one. Finally, we show the similarity between [CLS] and [MEAN] in Appendix C.

4.2 Linear Separability of Representations

In Figure 1(b), we discover the gap of improvement between the visualization effect and the performance, which may be due to the limitations of visualization technology. This leads us to introduce another method to quantify the quality of representations. Therefore, we apply a probing technique based on the idea of clustering—DIRECTPROBE to study the geometric structure of representations.

We select five common pre-trained models to examine the geometric structure of representations after fine-tuning. Table 1 shows the results that contain base and large versions corresponding to selected models. We discover that the better performance the model achieves, the fewer clusters the representation is divided into, i.e., the representation has higher linear separability. In particular, there are some models whose representations are divided into two and three clusters on HANS and MNLI, respectively (i.e., equaling the number of label categories), which suggests that all examples with the same label are in one cluster and there are no overlaps between each cluster. However, these

	MNLI		HANS	
Models	#clusters	Acc	#clusters	Acc
$\text{BERT}_{\text{base}}$	27	84.25	4	65.01
$\text{RoBERTa}_{\text{base}}$	5	88.10	3	69.53
$\text{DeBERTa}_{\text{base}}$	4	88.75	2	76.61
$\text{BART}_{\text{distill}}$	3	89.56	2	67.37
$\text{ELECTRA}_{\text{base}}$	4	88.77	2	76.84
$\text{BERT}_{\text{large}}$	5	86.69	2	68.77
$\text{RoBERTa}_{\text{large}}$	3	90.60	2	73.73
$\text{DeBERTa}_{\text{large}}$	3	91.28	2	78.07
$\text{BART}_{\text{large}}$	3	90.16	2	72.88
$\text{ELECTRA}_{\text{large}}$	3	90.47	2	78.21

Table 1: The number of clusters and corresponding accuracy from selected pre-trained language models on MNLI and HANS. There is an around 20% generalization gap between MNLI and HANS.

models achieve about 90% accuracy on MNLI but only no more than 80% accuracy on HANS, which is not consistent with the linear separability of the representation. Thus, we assume that **the low robustness of the NLU models is not due to the inseparability of representations.**

4.3 Similarity of Representations

In §4.1 and §4.2, we analyze representations with the t-SNE and DIRECTPROBE technique, respectively. However, what puzzles us is why the representation is linearly separable, while the performance of PLMs is not perfect. To study that, we investigate the cosine similarity of representations to find the reason for the flawed performance. In practice, we investigate the similarity between cluster centers within MNLI or HANS and between

Models	M-E|H-E	M-E|H-N	H-E|H-N	Acc
$\text{BERT}_{\text{base}}$	0.9376	0.8128	0.8597	65.01
$\text{RoBERTa}_{\text{base}}$	0.9562	0.8714	0.8131	69.53
$\text{DeBERTa}_{\text{base}}$	0.9733	0.7196	0.6921	76.61
$\text{BART}_{\text{distill}}$	0.9138	0.8067	0.9064	67.37
$\text{ELECTRA}_{\text{base}}$	0.9656	0.6550	0.6911	76.84
$\text{BERT}_{\text{large}}$	0.9573	0.7859	0.8511	68.77
$\text{RoBERTa}_{\text{large}}$	0.9286	0.7451	0.6810	73.73
$\text{DeBERTa}_{\text{large}}$	0.9545	0.6721	0.6912	78.07
$\text{BART}_{\text{large}}$	0.9145	0.7126	0.8467	72.88
$\text{ELECTRA}_{\text{large}}$	0.9560	0.5650	0.5891	78.21

Table 2: The similarity of [CLS] between two centers of selected embeddings and the accuracy on HANS. M-E indicates MNLI-Entailment; H-E indicates HANS-Entailment; H-N indicates HANS-Not-Entailment. It suggests that the representations with opposite semantics are similar in the semantic representation space.

MNLI and HANS. Each cluster includes all [CLS] embeddings with the same label on one dataset. The cluster center is defined as the average of all [CLS] embeddings in the cluster.

First, we compute the cosine similarity between MNLI-Entailment and HANS-Entailment/HANS-Not-Entailment as shown in Table 2. Ideally, the cosine similarity between MNLI-Entailment and HANS-Entailment is close to +1, and the cosine similarity between MNLI-Entailment and HANS-Not-Entailment is close to -1. However, the latter is not supported by our experiments, as shown in Table 2. It is even larger than +0.5, suggesting that the representations of HANS-Entailment and HANS-Not-Entailment are similar to those of MNLI-Entailment in the semantic representation space. Then, we compute the cosine similarity between HANS-Entailment and HANS-Not-Entailment as Table 2 shows, which should be close to -1. In fact, it is large than +0.5, suggesting that the representations of HANS-Entailment and HANS-Not-Entailment are similar, despite the representations from most of the selected models are linearly separable as Table 1 shows. These results mean that the encoders fail to distinguish Not-Entailment examples where the heuristics fail from Entailment examples well. In Appendix D, the gap of performance between Entailment and Not-Entailment on HANS confirm that.

Motivated by the above finding, we compute the Spearman correlation coefficient between similarity and accuracy and find that the accuracy is significantly inversely associated with the similarity between MNLI/HANS-Entailment and HANS-Not-Entailment. The correlation coefficients are -0.8788 and -0.8909 with a p-value less than 0.05. We suppose that **the significant similarity in semantic representation between MNLI/HANS-Entailment and HANS-Not-Entailment is the main reason why it is difficult to improve the performance on HANS.** Finally, we show the other similarity between cluster centers with different labels in Appendix E.

4.4 Analysis of Debiasing Methods

Typically, the poor performance of models on OOD examples is attributed to the fact that models only capture shortcut features (i.e., spurious features) but not robustness features (i.e. task-relevant features). Consequently, many debiasing methods are proposed to make models pay more attention to robustness features to improve the performance on OOD examples. In this work, we make detailed analyses of the impact of debiasing methods on fine-tuning models. For pre-trained language models, we select BERT and RoBERTa.

Based on the fact that the NLI task is considered as a classification task, the models for the NLI task can be divided into encoders and classifiers. Generally speaking, the encoder is from PLMs, and the classifier is the MLP network. Based on this architecture of the NLI models, we intend to study how debiasing methods work on encoders and classifiers, respectively. In practice, we apply a two-phase training strategy. Specifically, we first fine-tune models consisting of encoders and classifiers, then fix encoders and only retrain classifiers.

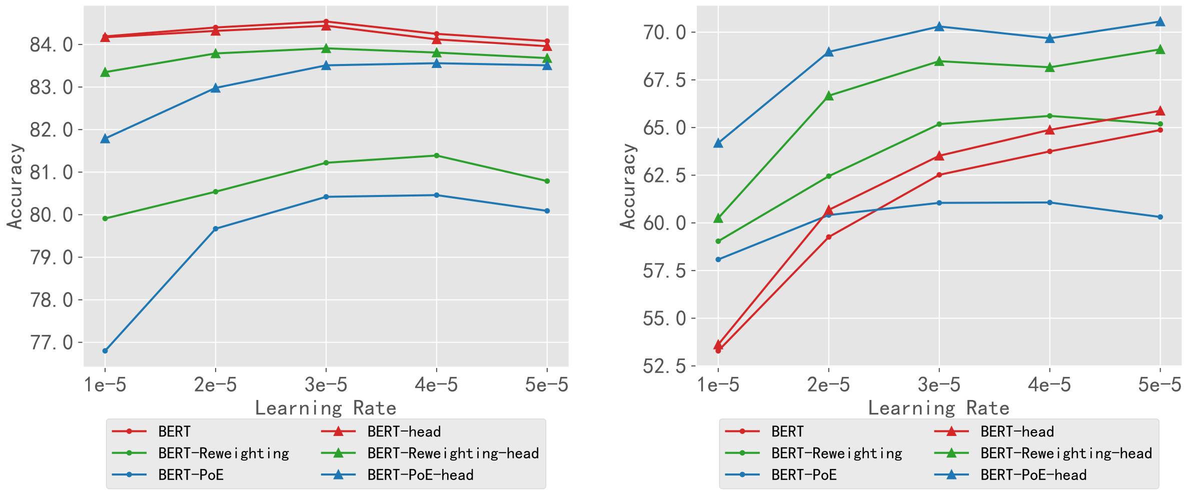
Figures 3 and 4 show the results on MNLI and HANS for BERT and RoBERTa, respectively. We take the learning rate from $1 * 10^{-5}$ to $5 * 10^{-5}$ to investigate the effect of the learning rate on the convergence of models. We discover that by fixing the encoder of fine-tuned models without debiasing methods (i.e. raw fine-tuned models) and only retraining the classifier with debiasing methods, the performance of models is improved on HANS for BERT and RoBERTa. Especially for BERT, the retraining strategy with debiasing methods achieves better performance than fine-tuning the whole model using debiasing methods both on MNLI and on HANS. Thus, we assume that **debiasing methods distort the representation to some degree** for BERT, which is similar to the finding of Mendelson and Belinkov (2021). Meanwhile, this strategy mitigates the degradation of performance on MNLI and further improves the performance on HANS. Noting that retraining the classifiers of

Models	Learning Rate				
	1e-5	2e-5	3e-5	4e-5	5e-5
BERT-ReW	79.91	80.54	81.22	81.39	80.79
BERT-ReW-head-self	79.92	80.61	81.29	81.39	80.90
BERT-PoE	76.80	79.67	80.42	80.46	80.09
BERT-PoE-head-self	77.06	79.90	80.58	80.61	80.32
RoBERTa-ReW	85.04	85.49	85.33	84.90	84.52
RoBERTa-ReW-head-self	85.18	85.41	85.29	84.77	84.44
RoBERTa-PoE	84.20	84.63	84.80	84.66	84.07
RoBERTa-PoE-head-self	84.21	84.78	84.97	84.60	84.14

Table 3: The results of the model trained with the debiasing method and retraining the classifier of that with the same debiasing method for MNLI. There is no obvious change on performance before and after retraining.

Models	Learning Rate				
	1e-5	2e-5	3e-5	4e-5	5e-5
BERT-ReW	59.04	62.45	65.18	65.61	65.19
BERT-ReW-head-self	59.35	62.70	65.17	65.70	65.31
BERT-PoE	58.08	60.41	61.05	61.07	60.31
BERT-PoE-head-self	58.01	60.12	60.95	61.02	60.10
RoBERTa-ReW	75.66	77.69	77.74	76.40	75.90
RoBERTa-ReW-head-self	75.75	77.72	77.70	76.39	75.82
RoBERTa-PoE	77.29	78.50	78.29	77.64	76.27
RoBERTa-PoE-head-self	77.04	78.37	78.17	77.72	76.19

Table 4: The results of the model trained with the debiasing method and retraining the classifier of that with the same debiasing method for HANS. There is no obvious change on performance before and after retraining.



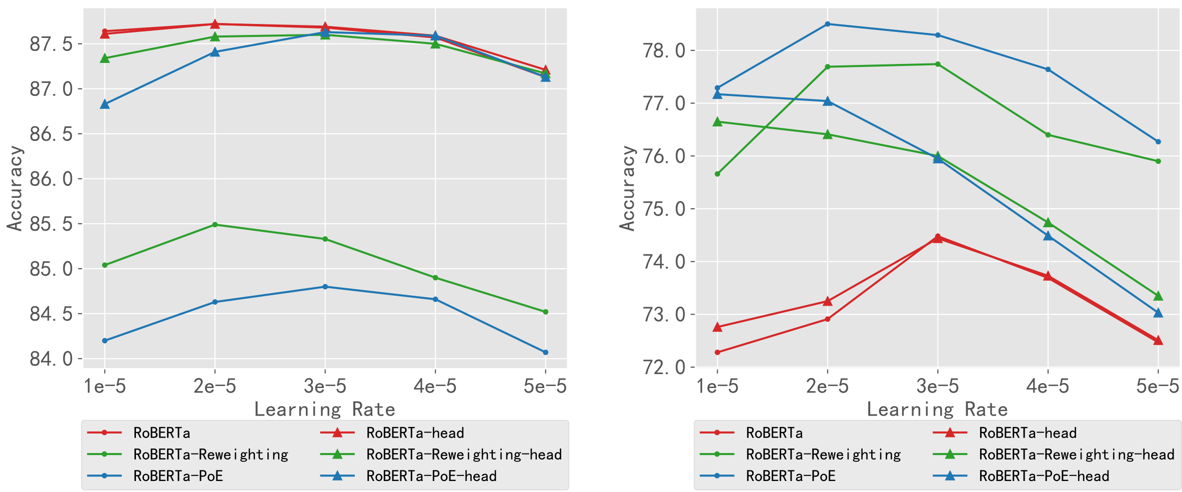
(a) BERT-MNLI

(b) BERT-HANS

Figure 3: (a) and (b) indicate the results of BERT on MNLI and HANS, respectively. The lines with dots: fine-tune models with or without debiasing methods. The lines with triangles: retrain classifiers using encoders from fine-tuned models with or without debiasing methods.

raw fine-tuned models without debiasing methods does not achieve an obvious effect on the performance, we find **classifiers play a significant role in the shortcut learning behavior of PLMs**. We design the comparative experiments to further clarify whether the performance improvement derives from the decoupled training strategy. Tables 3 and 4 show the results for MNLI and HANS, respectively. We discover that the decoupled training strategy also does not achieve an obvious effect for fine-tuned models with debiasing methods. This suggests that the performance improvement is derived from not the decoupled training strategy but the classifiers optimized by debiasing methods, and that the models fine-tuned with debiasing methods are not limited in performance by the classifiers.

Based on the above results, we suppose that a decoupled retraining strategy with debiasing methods can be considered as a probing tool, which is used to measure the shortcut learning behavior of PLMs from representations or classifiers. The performance gap between the raw fine-tuned model and the model with the classifier retrained using the debiasing methods measures the extent of shortcut learning behavior from classifiers. The gap of the performance between the model with the classifier retrained using the debiasing methods and the whole fine-tuned model with the debiasing methods measures the extent of shortcut learning behavior from representations. Through this probing tool, we can better understand how representations and classifiers affect the robustness of models.



(a) RoBERTa-MNLI

(b) RoBERTa-HANS

Figure 4: (a) and (b) indicate the results of RoBERTa on MNLI and HANS, respectively. The lines with dots: fine-tune models with or without debiasing methods. The lines with triangles: retrain classifiers using encoders from fine-tuned models with or without debiasing methods.

5 Related Work

Recently, the shortcut learning behavior for the language task is revealed in previous work (Niven and Kao, 2019; Mudrakarta et al., 2018; Geirhos et al., 2020). For the NLI task, the shortcut learning behavior in models is often investigated using challenge datasets (Jia and Liang, 2017; Naik et al., 2018; Glockner et al., 2018; McCoy et al., 2019). To mitigate this behavior, we can use advanced pre-trained language models to obtain better representations (Liu et al., 2019; Lewis et al., 2020; Clark et al., 2020b; He et al., 2021), or apply debiasing methods to fine-tune language models (Schuster et al., 2019; Clark et al., 2019; Utama et al., 2020b; Utama et al., 2020a).

There are lots of works that analyze and understand learned representations with probing techniques. For instance, Tenney et al. (2019), Hewitt et al. (2021) and Whitney et al. (2021) consider classifiers as probes. Meanwhile, Mimno and Thompson (2017), Ethayarajh (2019) and Zhou and Srikumar (2021) inspect the representations from a geometric perspective. There are also efforts to understand pre-trained representations (Chen et al., 2021; Li et al., 2021) and fine-tuned ones (Zhou and Srikumar, 2022) respectively. In contrast, we focus our analysis on biased features and classifiers, and study the role that the quality of representations and the capability of classifiers play in the robustness of models respectively.

6 Conclusion

In this work, we conduct an empirical study on how the robustness of language models is affected by encoders and classifiers, respectively.

On the one hand, we show that *the low robustness of language models is not primarily due to representations not being easily separable*. i) We find that several excellent models provide linearly separable representations, which suggests that classifiers limit the performance of models. ii) We find that the significantly high similarity between representations with opposite semantics from in-distribution and out-of-distribution datasets is a reason for the low robustness.

On the other hand, we show *the relative role of representations and classifiers in the low robustness of language models*. i) We find that debiasing methods do not always improve the quality of representations but rather improve the performance of models with optimal classifiers. ii) We find that the robustness of models depends not only on the low quality of representations, but also on the capability of classifiers, and their ratios vary for different architectures and fine-tuning processes.

Finally, we hope that the insights obtained from the empirical analysis will be beneficial to the community, allowing them to pay more attention to the important roles of classifiers for models and design better solutions to alleviate shortcut learning and improve the robustness of PLMs in NLU tasks.

Limitations

Despite our findings that both representations and classifiers affect the robustness of models, we are not successful in making use of that to further improve the understanding of models for the language. As a result, we plan to further research advanced methods that are capable of optimizing encoders and classifiers, respectively. Furthermore, the designed experiments in our analysis focus only on the NLI task in NLU tasks. Given the similarity between the NLU tasks, it may be possible to extrapolate the corresponding findings to other NLU tasks. In the future, we will consider the following NLU tasks and datasets: IMDB (Maas et al., 2011) / Yelp (Zhang et al., 2015) for the sentiment classification task; QQP (Iyer et al., 2017) / TwitterPPDB(TPPDB) (Lan et al., 2017) for the paraphrase identification task; FEVER (Thorne et al., 2018) / FeverSymmetric (Schuster et al., 2019) for the fact verification task.

Ethics Statement

This paper does not raise ethical concerns. This study does not involve any human subjects, practices to data set releases, potentially harmful insights, discrimination/bias/fairness concerns and privacy and security issues.

References

Yonatan Belinkov. 2021. Probing classifiers: Promises, shortcomings, and alternatives. *arXiv e-prints*, pages arXiv–2102.

Boli Chen, Yao Fu, Guangwei Xu, Pengjun Xie, Chuanqi Tan, Mosha Chen, and Liping Jing. 2021. Probing {bert} in hyperbolic spaces. In *International Conference on Learning Representations*.

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. 2020a. 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.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020b. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

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.

Mengnan Du, Varun Manjunatha, Rajiv Jain, Ruchi Deshpande, Franck Dernoncourt, Jiuxiang Gu, Tong Sun, and Xia Hu. 2021. Towards interpreting and mitigating shortcut learning behavior of NLU models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 915–929, Online. Association for Computational Linguistics.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. 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 55–65, Hong Kong, China. Association for Computational Linguistics.

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

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

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

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

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

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

John Hewitt, Kawin Ethayarajh, Percy Liang, and Christopher Manning. 2021. Conditional probing: measuring usable information beyond a baseline. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1626–1639, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shankar Iyer, Nikhil Dandekar, and Kornél Csernai. 2017. Quora question pairs. *First Quora Dataset Release: Question Pairs*.

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

Rabeeh Karimi Mahabadi, Yonatan Belinkov, and James Henderson. 2020. End-to-end bias mitigation by modelling biases in corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8706–8716, Online. Association for Computational Linguistics.

Katarzyna Krasnowska-Kieraś and Alina Wróblewska. 2019. Empirical linguistic study of sentence embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5729–5739, Florence, Italy. Association for Computational Linguistics.

Wuwei Lan, Siyu Qiu, Hua He, and Wei Xu. 2017. A continuously growing dataset of sentential paraphrases. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1224–1234, Copenhagen, Denmark. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Bai Li, Zining Zhu, Guillaume Thomas, Yang Xu, and Frank Rudzicz. 2021. How is BERT surprised? layerwise detection of linguistic anomalies. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4215–4228, Online. 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. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

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

Michael Mendelson and Yonatan Belinkov. 2021. Debiasing methods in natural language understanding make bias more accessible. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1545–1557, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

David Mimno and Laure Thompson. 2017. The strange geometry of skip-gram with negative sampling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2873–2878, Copenhagen, Denmark. Association for Computational Linguistics.

Pramod Kaushik Mudrakarta, Ankur Taly, Mukund Sundararajan, and Kedar Dhamdhere. 2018. Did the model understand the question? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1896–1906, Melbourne, Australia. Association for Computational Linguistics.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Timothy Niven and Hung-Yu Kao. 2019. Probing neural network comprehension of natural language arguments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4658–4664, Florence, Italy. Association for Computational Linguistics.

Christian S Perone, Roberto Silveira, and Thomas S Paula. 2018. Evaluation of sentence embeddings in downstream and linguistic probing tasks. *arXiv preprint arXiv:1806.06259*.

Yada Pruksachatkun, Jason Phang, Haokun Liu, Phu Mon Htut, Xiaoyi Zhang, Richard Yuanzhe Pang, Clara Vania, Katharina Kann, and Samuel R. Bowman. 2020. Intermediate-task transfer learning with pretrained language models: When and why does it work? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5231–5247, Online. Association for Computational Linguistics.

Ivan Sanchez, Jeff Mitchell, and Sebastian Riedel. 2018. Behavior analysis of NLI models: Uncovering the influence of three factors on robustness. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1975–1985, New Orleans, Louisiana. Association for Computational Linguistics.

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

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

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Sam Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What do you learn from context? probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018. The fact extraction and VERification (FEVER) shared task. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 1–9, Brussels, Belgium. Association for Computational Linguistics.

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.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

William F Whitney, Min Jae Song, David Brandfonbrener, Jaan Altosaar, and Kyunghyun Cho. 2021. Evaluating representations by the complexity of learning low-loss predictors. In *Neural Compression: From Information Theory to Applications – Workshop @ ICLR 2021*.

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

Ruibin Xiong, Yimeng Chen, Liang Pang, Xueqi Cheng, Zhi-Ming Ma, and Yanyan Lan. 2021. Uncertainty calibration for ensemble-based debiasing methods. *Advances in Neural Information Processing Systems*, 34.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Yichu Zhou and Vivek Srikumar. 2021. DirectProbe: Studying representations without classifiers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5070–5083, Online. Association for Computational Linguistics.

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

A Implementation Details

Only-bias Model The only-bias model is trained on random 2,000 of examples for 3 epochs. We use AdamW optimizer with the default learning rate $5*10^{-5}$, where the betas are set as [0.9, 0.999] and the L2 weight decay is set to 0.01. The batch size is set to 32, and the warmup ratio is set to 0.1.

Main Model The hidden dimension of the classifiers is the same as the output of encoders (i.e., base version: 768; large version: 1024). The activation functions of the classifiers are the same as the setup of the Huggingface Transformers (i.e., Tanh or GELU (Hendrycks and Gimpel, 2016)): i) BERT, BART, and RoBERTa are Tanh; ii) DeBERTa and ELECTRA are GELU. The parameters of PLMs are shown as Table 5.

Retrain Classifiers with or without Debiasing Methods The parameters of the classifier are initialized by a normal distribution with the mean of 0.0 and the variance of 0.02. We use AdamW optimizer with the default learning rate $5*10^{-5}$, where the betas are set to [0.9, 0.999] and the L2 weight decay is set to 0.01. The batch size is set to 32 and the warmup ratio is set to 0.1. We retrain the classifiers for 3 epochs.

Models	Parameters
$\mathrm{BERT}_{\mathrm{base}}$	110M
$\mathrm{BERT}_{\mathrm{large}}$	340M
$\mathrm{RoBERTa}_{\mathrm{base}}$	125M
$\mathrm{RoBERTa}_{\mathrm{large}}$	355M
$\mathrm{DeBERTa}_{\mathrm{base}}$	134M
$\mathrm{DeBERTa}_{\mathrm{large}}$	390M
$\mathrm{BART}_{\mathrm{distill}}$	356M
$\mathrm{BART}_{\mathrm{large}}$	406M
$\mathrm{ELECTRA}_{\mathrm{base}}$	110M
$\mathrm{ELECTRA}_{\mathrm{large}}$	335M

Table 5: The parameters of pre-trained language models.

B Other Results of Visualization

Figures 5 and 6 show the other visualization results for [CLS] and [MEAN], respectively.

C Similarity between [CLS] and [MEAN]

To explore how [CLS] and [MEAN] are related in terms of robustness, we compute the cosine similarity between [CLS] and [MEAN] on MNLI and HANS, respectively. Table 6 summarizes the results. We compare the change in similarity from base models to large ones and discover that the changes in similarity have different trends. When the trend of the change in similarity increases, we suppose that the model is likely to learn similar information. On the contrary, the model is likely to learn different information. Based on this observation, we conjecture that it is possible to improve the robustness of models by figuring out how the amount of information learned affects performance and introducing the information from [MEAN] as supervised signals while fine-tuning. Verifying or rejecting this conjecture requires further study.

	MNLI		HANS	
Models	Similarity	Acc	Similarity	Acc
$\mathrm{BERT}_{\mathrm{base}}$	0.7683	84.25	0.7441	65.01
$\mathrm{BERT}_{\mathrm{large}}$	0.6364	86.69	0.6146	68.77
$\mathrm{RoBERTa}_{\mathrm{base}}$	0.8224	88.10	0.7663	69.53
$\mathrm{RoBERTa}_{\mathrm{large}}$	0.9845	90.60	0.9914	73.73
$\mathrm{DeBERTa}_{\mathrm{base}}$	0.5718	88.75	0.5690	76.61
$\mathrm{DeBERTa}_{\mathrm{large}}$	0.3362	91.28	0.2371	78.07
$\mathrm{BART}_{\mathrm{distill}}$	0.6375	89.56	0.6903	67.37
$\mathrm{BART}_{\mathrm{large}}$	0.5989	90.16	0.6433	72.88
$\mathrm{ELECTRA}_{\mathrm{base}}$	0.5188	88.77	0.6048	76.84
$\mathrm{ELECTRA}_{\mathrm{large}}$	0.8461	90.47	0.9317	78.21

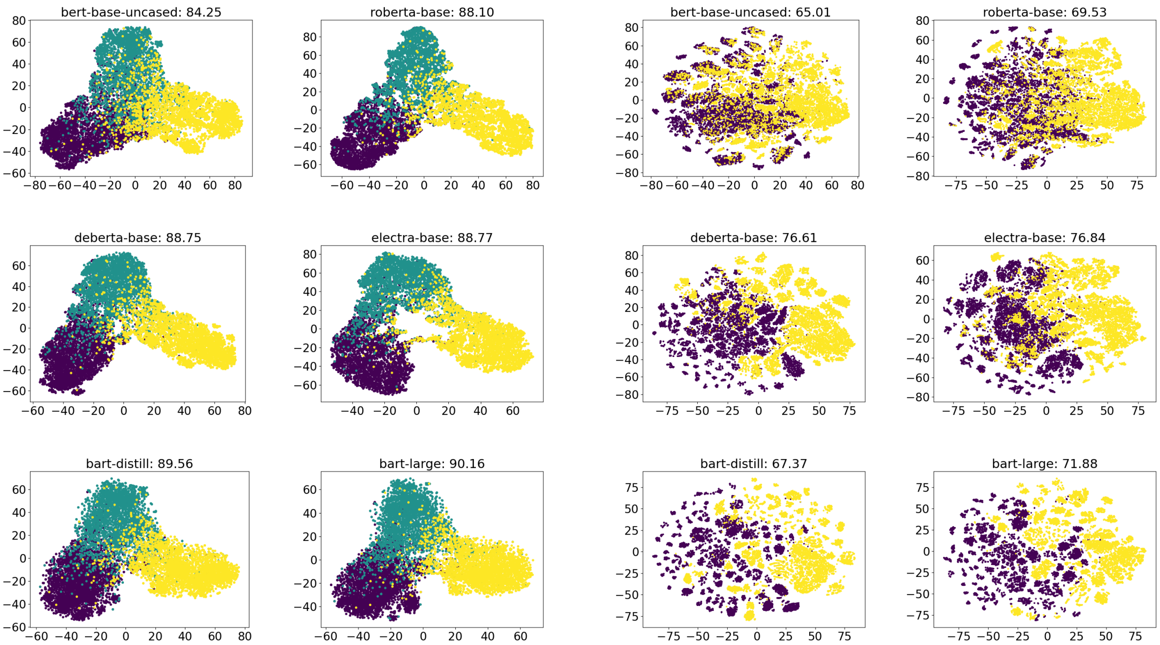
Table 6: The cosine similarity between [CLS] and [MEAN] and corresponding accuracy from selected pre-trained language models on MNLI and HANS.

	MNLI		HANS	
Models	[CLS]	[MEAN]	[CLS]	[MEAN]
$\mathrm{BERT}_{\mathrm{base}}$	84.25	84.17	65.01	65.20
$\mathrm{RoBERTa}_{\mathrm{base}}$	88.10	87.94	69.53	70.89
$\mathrm{DeBERTa}_{\mathrm{base}}$	88.75	88.89	76.61	78.12
$\mathrm{BART}_{\mathrm{distill}}$	89.56	88.86	67.37	63.20
$\mathrm{ELECTRA}_{\mathrm{base}}$	88.77	88.47	76.84	76.21
$\mathrm{BERT}_{\mathrm{large}}$	86.69	85.35	68.77	67.57
$\mathrm{RoBERTa}_{\mathrm{large}}$	90.60	90.52	73.73	74.82
$\mathrm{DeBERTa}_{\mathrm{large}}$	91.28	91.34	78.07	78.54
$\mathrm{BART}_{\mathrm{large}}$	90.16	88.95	71.88	71.34
$\mathrm{ELECTRA}_{\mathrm{large}}$	90.47	90.50	78.21	78.16

Table 7: The results of [CLS] and [MEAN] for MNLI and HANS.

D Results of HANS in detail

Table 10 shows the results of HANS in detail.



(a) [CLS] in MNLI (b) [CLS] in HANS

Figure 5: The t-SNE visualization result of [CLS] embeddings from different pre-trained language models on MNLI and HANS. The words above figures are the selected model and corresponding accuracy.

	Learning Rate				
Models	1e-5	2e-5	3e-5	4e-5	5e-5
BERT	84.19	84.40	84.54	84.25	84.08
BERT-ReW	79.91	80.54	81.22	81.39	80.79
BERT-PoE	76.80	79.67	80.42	80.46	80.09
BERT-head	84.17	84.32	84.44	84.12	83.96
BERT-ReW-head	83.35	83.79	83.91	83.81	83.68
BERT-PoE-head	81.79	82.98	83.51	83.56	83.51
BERT-ReW-head-self	79.92	80.61	81.29	81.39	80.90
BERT-PoE-head-self	77.06	79.90	80.58	80.61	80.32
RoBERTa	87.64	87.72	87.68	87.57	87.13
RoBERTa-ReW	85.04	85.49	85.33	84.90	84.52
RoBERTa-PoE	84.20	84.63	84.80	84.66	84.07
RoBERTa-head	87.61	87.72	87.69	87.59	87.21
RoBERTa-ReW-head	87.34	87.58	87.60	87.50	87.17
RoBERTa-PoE-head	86.83	87.41	87.63	87.59	87.13
RoBERTa-ReW-head-self	85.18	85.41	85.29	84.77	84.44
RoBERTa-PoE-head-self	84.21	84.78	84.97	84.60	84.14

Table 8: The complete results for MNLI.

E Similarity between Cluster Centers

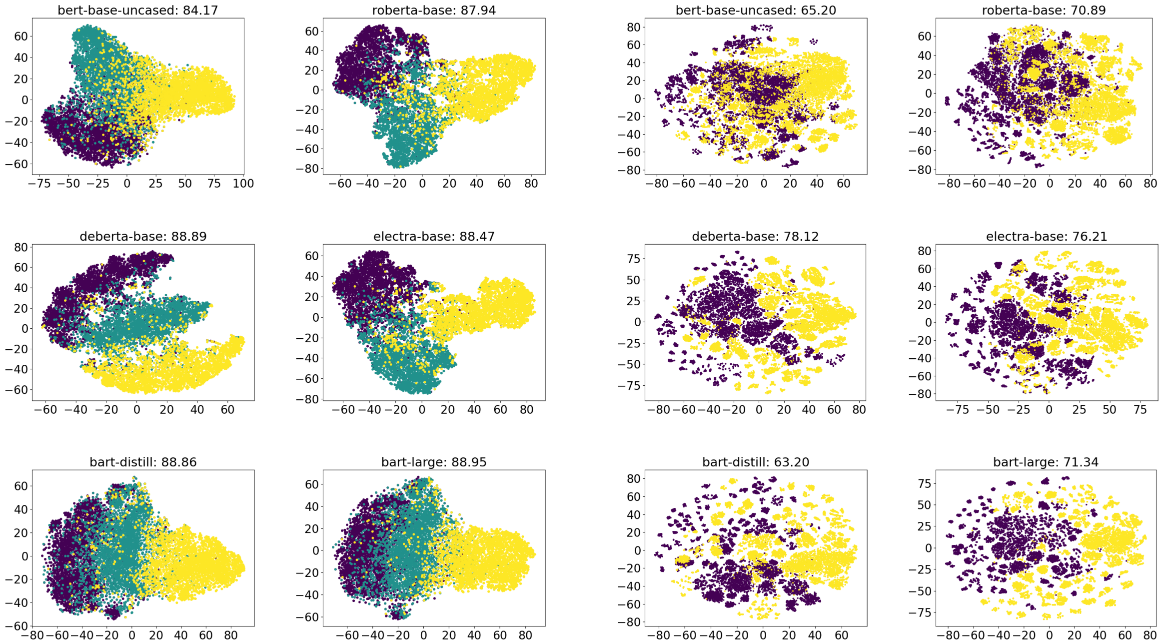
Table 11 shows the complete similarity between cluster centers within MNLI or HANS and between MNLI and HANS.

F Complete Results

Tables 8 and 9 show the complete results of BERT and RoBERTa for MNLI and HANS, respectively.

	Learning Rate				
Models	1e-5	2e-5	3e-5	4e-5	5e-5
BERT	53.29	59.26	62.52	63.75	64.87
BERT-ReW	59.04	62.45	65.18	65.61	65.19
BERT-PoE	58.08	60.41	61.05	61.07	60.31
BERT-head	53.62	60.68	63.52	64.88	65.88
BERT-ReW-head	60.25	66.67	68.48	68.16	69.10
BERT-PoE-head	64.19	68.97	70.30	69.68	70.56
BERT-ReW-head-self	59.35	62.70	65.17	65.70	65.31
BERT-PoE-head-self	58.01	60.12	60.95	61.02	60.10
RoBERTa	72.28	72.91	74.48	73.69	72.47
RoBERTa-ReW	75.66	77.69	77.74	76.40	75.90
RoBERTa-PoE	77.29	78.50	78.29	77.64	76.27
RoBERTa-head	72.76	73.25	74.44	73.73	72.51
RoBERTa-ReW-head	76.65	76.41	76.00	74.74	73.35
RoBERTa-PoE-head	77.17	77.04	75.95	74.49	73.03
RoBERTa-ReW-head-self	75.75	77.72	77.70	76.39	75.82
RoBERTa-PoE-head-self	77.04	78.37	78.17	77.72	76.19

Table 9: The complete results for HANS.



(a) [MEAN] in MNLI (b) [MEAN] in HANS

Figure 6: The t-SNE visualization result of [MEAN] embeddings from different pre-trained language models on MNLI and HANS. The words above figures are the selected model and corresponding accuracy.

Models	HANS	HANS-Entailment	HANS-Not-Entailment	Entailment Category			Non-Entailment Category		
				Overlap	Subsequence	Constituent	Overlap	Subsequence	Constituent
$\text{BERT}_{\text{base}}$	65.01	98.97	31.05	97.54	99.64	99.74	59.10	12.12	21.94
$\text{RoBERTa}_{\text{base}}$	69.53	99.39	39.66	98.96	99.98	99.24	66.02	19.72	33.24
$\text{DeBERTa}_{\text{base}}$	76.61	99.21	54.01	97.82	100.0	99.80	95.60	30.70	35.72
$\text{BART}_{\text{distill}}$	67.37	99.25	35.49	98.32	99.72	99.70	69.18	16.36	20.94
$\text{ELECTRA}_{\text{base}}$	76.84	99.59	54.09	98.90	99.94	99.94	95.92	27.98	38.38
$\text{BERT}_{\text{large}}$	68.77	94.85	42.69	88.22	97.60	98.74	74.90	22.62	30.56
$\text{RoBERTa}_{\text{large}}$	73.73	99.63	47.83	99.98	100.00	98.92	90.52	34.82	18.14
$\text{DeBERTa}_{\text{large}}$	78.07	99.86	56.27	99.74	100.00	99.84	95.00	33.28	40.54
$\text{BART}_{\text{large}}$	71.88	99.53	44.24	99.02	99.76	99.80	80.76	27.32	24.64
$\text{ELECTRA}_{\text{large}}$	78.21	99.84	56.58	99.52	100.00	100.00	93.04	37.24	39.46

Table 10: The results of HANS in detail.

Models	M-E|H-E	M-E|H-N	M-N|H-E	M-N|H-N	M-C|H-E	M-C|H-N	M-E|M-N	M-E|M-C	M-N|M-C	H-E|H-N	Acc
$\text{BERT}_{\text{base}}$	0.9376	0.8128	-0.0725	0.0470	-0.1906	0.2225	0.1744	-0.2330	0.1160	0.8597	65.01
$\text{RoBERTa}_{\text{base}}$	0.9562	0.8714	0.1196	0.3833	-0.1000	0.3951	0.3640	-0.0354	0.2031	0.8131	69.53
$\text{DeBERTa}_{\text{base}}$	0.9733	0.7196	0.1833	0.4205	-0.1263	0.4709	0.2963	-0.1289	0.1063	0.6921	76.61
$\text{BART}_{\text{distill}}$	0.9138	0.8067	0.1695	0.2850	0.1791	0.4585	0.3210	0.1178	0.2796	0.9064	67.37
$\text{ELECTRA}_{\text{base}}$	0.9656	0.6550	-0.1790	-0.0977	-0.3068	0.3648	0.0150	-0.3798	-0.1271	0.6911	76.84
$\text{BERT}_{\text{large}}$	0.9573	0.7859	0.0926	0.3132	0.1037	0.4651	0.2327	0.0019	0.2867	0.8511	68.77
$\text{RoBERTa}_{\text{large}}$	0.9286	0.7451	-0.1081	0.0972	-0.0796	0.4204	0.1048	-0.1673	-0.1164	0.6810	73.73
$\text{DeBERTa}_{\text{large}}$	0.9545	0.6721	0.1204	0.3466	-0.0546	0.5147	0.2880	-0.1025	0.0611	0.6912	78.07
$\text{BART}_{\text{large}}$	0.9145	0.7126	0.0698	0.1178	0.1050	0.4824	0.1828	-0.0084	0.0782	0.8467	72.88
$\text{ELECTRA}_{\text{large}}$	0.9560	0.5650	-0.3017	-0.1654	-0.1751	0.5167	-0.1574	-0.2974	-0.2505	0.5891	78.21

Table 11: The similarity of [CLS] between two centers of selected embeddings and the accuracy on HANS. M-E indicates MNLI-Entailment; M-N indicates MNLI-Neutral; M-C indicates MNLI-Contradiction; H-E indicates HANS-Entailment; H-N indicates HANS-Not-Entailment.