Artificial Intelligence 339 (2025) 104258

Contents lists available at ScienceDirect

Artificial Intelligence

journal homepage: www.elsevier.com/locate/artint

A simple yet effective self-debiasing framework for transformer models

Xiaoyue Wang [a,1], Xin Liu [b,1], Lijie Wang [c], Suhang Wu [a], Jinsong Su [a,*], Hua Wu [b]

[a] School of Informatics, Xiamen University, Xiamen 361005, China
[b] University of Michigan, Ann Arbor 48109, USA
[c] Baidu Inc., Beijing 100085, China

ARTICLE INFO

Keywords:
Natural language understanding
Dataset bias
Self-debiasing

ABSTRACT

Current Transformer-based natural language understanding (NLU) models heavily rely on dataset biases, while failing to handle real-world out-of-distribution (OOD) instances. Many methods have been proposed to deal with this issue, but they ignore the fact that the features learned in different layers of Transformer-based NLU models are different. In this paper, we first conduct preliminary studies to obtain two conclusions: 1) both low- and high-layer sentence representations encode common biased features during training; 2) the low-layer sentence representations encode fewer unbiased features than the high-layer ones. Based on these conclusions, we propose a simple yet effective self-debiasing framework for Transformer-based NLU models. Concretely, we first stack a classifier on a selected low layer. Then, we introduce a residual connection that feeds the low-layer sentence representation to the top-layer classifier. In this way, the top-layer sentence representation will be trained to ignore the common biased features encoded by the low-layer sentence representation and focus on task-relevant unbiased features. During inference, we remove the residual connection and directly use the top-layer sentence representation to make predictions. Extensive experiments and in-depth analyses on NLU tasks demonstrate the superiority of our framework, achieving a new state-of-the-art (SOTA) on three OOD test sets.

1. Introduction

Recently, Transformer-based models have achieved competitive performance on various NLU benchmarks [1,2]. However, many studies show that these models tend to directly exploit biased features as shortcuts to make predictions without understanding the semantics of input texts [3–5]. As a result, this behavior leads to the low generalizability and poor robustness of these models on OOD instances [6]. For example, on the PAWS [7], which is the OOD test set for Quora Question Pairs (QQP) dataset.[2] The commonly-used BERT-based model [2] does not achieve the expected results, as shown in Table 1.

To deal with this issue, many model-agnostic debiasing methods have been proposed, which mainly involve two steps. The first step is identifying biased training instances, of which predictions are heavily influenced by biased features, via data analysis, researchers' task-specific insights [3,8–10] or bias-only models [11–15]. The second step is employing various methods to down-

* Corresponding author.
E-mail addresses: xiaoyuewang@stu.xmu.edu.cn (X. Wang), jssu@xmu.edu.cn (J. Su).

[1] Both authors contributed equally to this work.

[2] https://www.kaggle.com/c/quora-question-pairs.

https://doi.org/10.1016/j.artint.2024.104258
Received 11 February 2023; Received in revised form 21 April 2024; Accepted 21 November 2024

Available online 2 December 2024

Table 1
Two instances from PAWS [7]. Both instances contain biased features, which make the dominant model [2] unable to predict the relationship between their sentences correctly. In the first instance, its two sentences contain a high proportion of overlapping words, which convey different meanings. The second instance is a paraphrase sentence pair, while its two sentences contain a limited number of overlapping words.

Sentence 1: *" Captain " was broken up in 1762.* **Sentence 2**: *" Captain " was rolled up in 1762.* **Golden Label**: *non-duplicate* **Predicted Label**: *duplicate*
Sentence 1: *Is there a tutorial on how to use Quora?* **Sentence 2**: *How do I start using Quora?* **Golden Label**: *duplicate* **Predicted Label**: *non-duplicate*

weight the importance of biased training instances, such as example re-weighting [10,16], confidence regularization [11] and model ensemble [17,18,16].

Despite their success, most studies consider NLU models as black-box systems, ignoring that different layers of Transformer-based NLU model learn different features. As analyzed in previous studies [19,20], Transformer-based pre-trained language models are able to effectively capture rich linguistic knowledge, with surface features in low layers, syntactic features in middle layers, and semantic features in high layers. Thus, two questions naturally arise: 1) Are there differences in features learned in terms of bias by different layers, i.e., biased and unbiased feature learning? 2) If so, can we leverage these differences to alleviate biased feature learning?

To answer the first question, we conduct preliminary studies to explore feature learning in different layers of Transformer-based NLU models. Specifically, following Du et al. [21], we first identify biased and anti-biased training instances from the training set, and extract biased and anti-biased validation instances from the validation set. Then, we stack a classifier on the sentence representation of each Transformer layer. Afterwards, we analyze the feature learning of different layers from both model training and prediction perspectives. Experimental analyses show that 1) *the low- and high-layer sentence representations encode common biased features*, and 2) *the low-layer sentence representations encode fewer unbiased features than the high-layer ones*.

Based on the above analyses, we propose a self-debiasing framework for Transformer-based NLU models. Concretely, we first add a classifier on a selected low layer to encourage the low-layer sentence representation to encode more common biased features during training, which are also encoded in the high-layer classifier. Then, we introduce a residual connection [22] that feeds the low-layer sentence representation to the top-layer classifier. In this way, the top-layer sentence representation is encouraged to ignore the common biased features and pay attention to task-relevant unbiased features. Note that we remove the residual connection during inference and directly use the top-layer sentence representation to make predictions.

Finally, we conduct experiments on three NLU tasks. Experimental results show that our simple framework not only achieves better performance on the OOD test sets, but also maintains comparable performance on the validation sets, compared with previous methods [11,12,21]. Besides, we prove that our framework indeed improves the understanding ability of the model through in-depth analyses.

2. Related work

Our related works mainly include the studies on identifying biased instances and debiasing methods.

Identifying biased instances This task is crucial to the subsequent debiasing methods. In this respect, many researchers first manually characterize the specific types of dataset biases, including word co-occurrence [3,8–10] and lexico-syntactic patterns [23,24], and then identify biased instances according to these bias patterns. However, these methods heavily rely on researchers' intuition and task-specific insights, limiting their applications to various NLU tasks and datasets. To deal with this issue, some studies employ various methods to create bias-only models for identifying biased instances, such as using a tiny fraction of training data [12], partial inputs [25,26,16], or a simplified model architecture [13].

Debiasing methods There have been many attempts to reduce dataset biases through various data construction methods, such as adversarial filtering [27], human-in-the-loop [28] and controlled generation [29]. Despite their effectiveness, researchers also show that newly constructed datasets may not cover all biased patterns [30]. Therefore, many researchers resort to various robust algorithms based on their prior knowledge of task-specific biases. In this respect, some studies adopt adversarial learning to remove the hypothesis-only bias from NLI models. For example, Belinkov et al. [31] and Stacey et al. [32] apply the gradient reverse layer [33] to train an external classifier that forces the hypothesis encoder to ignore hypothesis-only biases. A complementary line of studies focuses on debiasing models by down-weighting the importance of biased instances during training, such as example re-weighting [10,12], confidence regularization [11], upweighting minority instances [34,35], and model ensemble [17,16]. Usually, these methods involve two models, i.e., a bias-only model used to identify biased instances and a robust model learning from unbiased instances. In addition to the above, very recently Lyu et al. [36] use contrastive learning to capture the dynamic influence of biases, then reduce biased features.

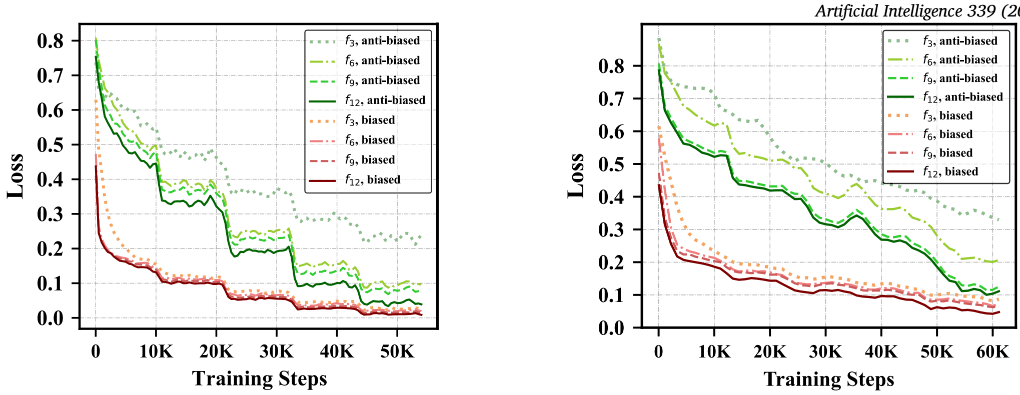


Fig. 1. Training loss curves of the BERT-based NLU model on biased and anti-biased training instances of QQP (a) and MNLI (b), where f_i represents the i-th classifier. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

Notably, most of previous studies consider models as black-box systems, and use the above two steps to debias models. By contrast, in this work, we explore the debiasing framework based on the internal structure of the model without manual analyses, extra bias-only models or complex hyper-parameter settings.

3. Feature learning in transformer models

In this work, we choose BERT [2] as our basic model, due to its competitive performance in many NLU tasks [1]. In this section, we first briefly introduce BERT, and then conduct preliminary studies to analyze the feature learning in different layers of the BERT-based NLU model.

3.1. Overview of BERT architecture

BERT stacks L identical layers, each containing a multi-head self-attention sub-layer, an MLP sub-layer, and a residual connection around these two sub-layers, followed by a layer normalization sub-layer.

Note that many studies on representation learning show that BERT can effectively capture rich linguistic knowledge, with different kinds of knowledge in different layers [19,20,37,19]. The following subsections aim to answer the two research questions shown in the Introduction.

3.2. Feature learning in different layers of transfomer models

To answer the above questions, we construct a BERT-based NLU model[3] and equip it with layer-specific classifiers based on sentence representations. Then, we analyze the features learning of these layers from both model training and prediction perspectives.

Previous studies [7,5] observed that the lexical overlap of two sentences is a typical biased feature in QQP, and a high lexical overlapping ratio usually co-occurs with some specific labels. Inspired by the above observation, we identify biased and anti-biased training instances from the QQP training set, and biased and anti-biased validation instances from the QQP validation set, respectively, based on the lexical overlapping ratio of each instance. Concretely, we first calculate the number of overlapping words and divide it by the maximum sentence length. Then, we identify an instance as a biased one if it satisfies the following: 1) its lexical overlapping ratio is greater than 70% and the label is "*duplicate*"; 2) it possesses a ratio less than 30% and is assigned with an "*non-duplicate*" label. Conversely, the instance with a ratio greater than 70% and "*non-duplicate*" label, or with a ratio less than 30% and "*duplicate*" label, is considered as an anti-biased instance.

Afterwards, we use the original QQP dataset to train the model and inspect the training losses of different classifiers on the biased and anti-biased training instances, respectively. From Fig. 1 (a),[4] we observe that all classifiers show similar trends in biased training instances. By contrast, the low-layer classifiers ($l = 3, 6$) possess higher training loss than the high-layer ones ($l = 9, 12$) on anti-biased training instances. Additionally, we conduct experiments on the MNLI dataset and employ the same strategy used for QQP to identify biased and anti-biased training instances. The results, presented in Fig. 1(b), show a trend similar to that observed in QQP, thereby reinforcing our findings.

Next, we compare the accuracies of classifiers on different validation instances of two datasets. From Fig. 2 (a) and (b), we can find that all classifiers exhibit similar performance on biased validation instances and suffer from performance degradation on anti-biased validation instances. Meanwhile, low-layer classifiers f_l $(1 \le l \le 5)$ perform worse than high-layer ones on anti-biased validation instances.

---

3 We also conduct preliminary experiments on RoBERTa and DeBERTa. Results can be found in Appendix B.

4 For the sake of clarity, here we only show the curves of four classifiers. In fact, other layer classifiers exhibit similar trends.

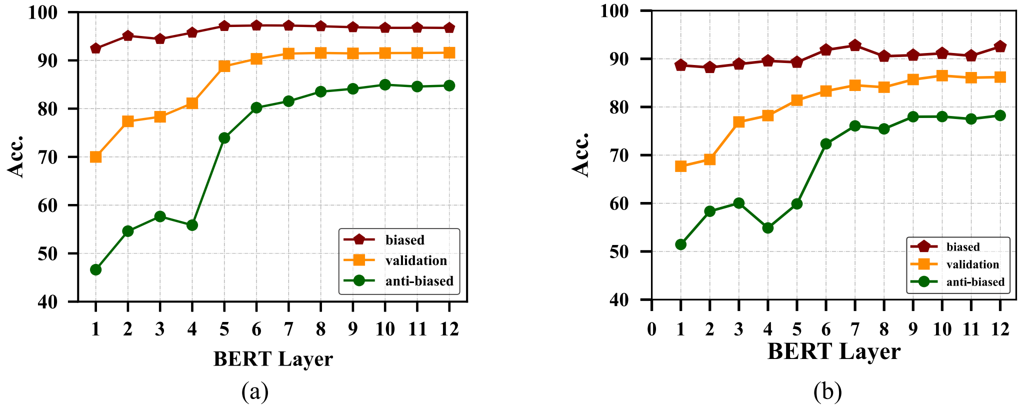


Fig. 2. The prediction performance of layer-specific classifiers of the BERT-based NLU model on the biased validation instances, the validation set, and the anti-biased validation instances of QQP (a) and MNLI (b) datasets. On the biased validation instances, low-layer classifiers f_l ($1 \le l \le 5$) perform slightly worse than high-layer ones, but much worse on anti-biased validation instances.

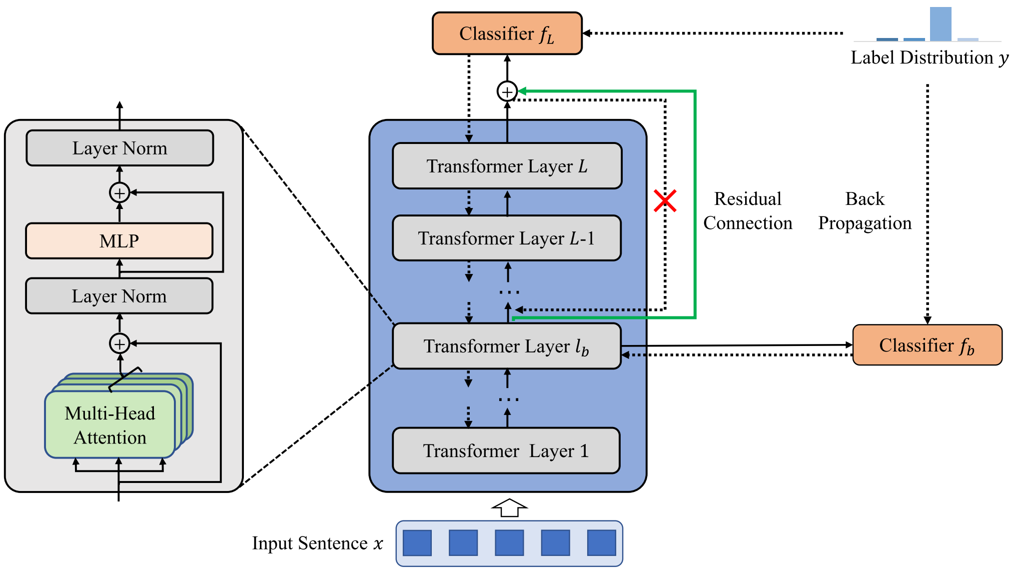


Fig. 3. Overview of our framework. In addition to the top-layer classifier f_L based on the top-layer sentence representation h^L_{CLS}, we select a low layer l_b and stack a classifier f_b on its sentence representation $h^{l_b}_{\text{CLS}}$. Then, we introduce a residual connection feeding the sum of $h^{l_b}_{\text{CLS}}$ and h^L_{CLS} to f_L. Through the model training, $h^{l_b}_{\text{CLS}}$ will encode common biased features, which have also been encoded in h^L_{CLS}, and thus h^L_{CLS} is encouraged to focus on unbiased features. Notably, we turn off the gradient calculation of the residual connection to avoid the f_L loss directly influence the representation learning of $h^{l_b}_{\text{CLS}}$. The green line denotes our introduced residual connection, and dash lines denote the backpropagation process.

Based on the above experimental results, we can draw the following conclusions: 1) The low- and high-layer sentence representations encode common biased features, which explains that low- and high-layer classifiers show similar loss trends on biased training instances and perform almost well on biased validation instances; 2) The low-layer sentence representations encode less useful task-relevant unbiased features than the high-layer ones so that their classifiers have higher losses on anti-biased training instances and obtain worse results on anti-biased validation instances.

4. Self-debiasing framework for transformer models

Based on the above analyses, we propose a framework for Transformer models by employing a residual connection to exploit the low-layer sentence representation to debias the top-layer sentence representation. Generally, it involves the following two steps.

Step1: low-layer sentence representation learning As shown in Fig. 3, given an input sentence x, we first employ a Transformer encoder to obtain the contextual representation for each token. Then, we select a low layer l_b and stack a classifier f_b on its sentence representation $h^{l_b}_{\text{CLS}}$, which we directly use the contextual representation of `[CLS]`. As analyzed in Section 3.2, $h^{l_b}_{\text{CLS}}$ will encode common biased features that would also be encoded by top-layer sentence representation h^L_{CLS} as the training of f_b goes on. Finally, we obtain the probability distribution p_b over labels as follows:

$$p_b = \mathrm{Softmax}(W_b h^{l_b}_{\mathrm{CLS}} + b_b), \tag{1}$$

where W_b and b_b are the learnable parameters. Here, we train the classifier f_b using the commonly-used cross-entropy loss:

$$\mathcal{L}_b = -\sum_{i=1}^{K} y^{(i)} \cdot \log(p_b^{(i)}), \tag{2}$$

where K denotes the number of labels, $y^{(i)}$ equals to 1 if the i-th label is the golden label, and 0 for other labels.

Step2: debiasing with a residual connection We introduce a residual connection [22] into our framework, which allows us to exploit the low-layer sentence representation $h^{l_b}_{\mathrm{CLS}}$ to debias the top-layer one h^L_{CLS}.

Specifically, through this residual connection, we use the sum of $h^{l_b}_{\mathrm{CLS}}$ and h^L_{CLS} as the input of the top-layer classifier f_L instead of h^L_{CLS}. Formally, the probability distribution output by f_L is calculated as follows:

$$p_L = \mathrm{Softmax}(W_L(h^{l_b}_{\mathrm{CLS}} + h^L_{\mathrm{CLS}}) + b_L), \tag{3}$$

where W_L and b_L are also trainable parameters. Note that f_L has the same architecture but different parameters with f_b, supervised by a cross-entropy loss:

$$\mathcal{L}_L = -\sum_{i=1}^{K} y^{(i)} \cdot \log(p_L^{(i)}). \tag{4}$$

The effectiveness of our design may be attributed to two factors: 1) As stated in Section 3.2, the low-layer representation contains fewer unbiased features than the high-layer ones. This indicates that h^L_{CLS} will encode the unbiased features that are not encoded in $h^{l_b}_{\mathrm{CLS}}$. 2) As $h^{l_b}_{\mathrm{CLS}}$ already encodes the common bias features, h^L_{CLS} is encouraged to ignore the common biased features which already encoded in $h^{l_b}_{\mathrm{CLS}}$.

Finally, the whole training objective is defined as follows:

$$\mathcal{L} = \mathcal{L}_b + \mathcal{L}_L. \tag{5}$$

Please notice that during training, we turn off the gradient calculation of the residual connection to remove the effect of $\mathcal{L}_L$ on the learning of $h^{l_b}_{\mathrm{CLS}}$. During inference, we remove $h^{l_b}_{\mathrm{CLS}}$ from Equation (3) and directly use h^L_{CLS} to make predictions.[5]

5. Experiments

5.1. Setup

Tasks and datasets We conduct several groups of experiments on three common NLU tasks: natural language inference, fact verification, and paraphrase identification. The datasets of each task contain a training set, a validation set, and its corresponding OOD test set.

- **Natural Language Inference** is to predict the entailment relationship between the pair of premise and hypothesis. We conduct experiments on MNLI [38] and SNLI [39] datasets using them as the ID set. We evaluate NLI models on their corresponding OOD test sets HANS [4] and the Scramble Test [40], respectively. These OOD test sets are specifically designed to assess whether models rely on syntactic and word-overlap biases to make predictions.
- **Fact Verification** aims to identify whether a claim is supported or refuted by the given evidence text. We adopt the FEVER dataset [41] as the ID set to train models, and assess the model abilities on the OOD test set—FeverSymmetric (Symm.) [10], which is created to reduce claim-only biases.
- **Paraphrase Identification** is to predict whether the given question pair is duplicate or non-duplicate in semantics. We use the QQP dataset as the ID set to train models, and evaluate model performance on the OOD test set—PAWS [7], which investigates whether the model exploits word overlapping to make predictions. Instances of OOD test sets are presented in Table 3. The basic statistics of all datasets used in our experiments are shown in Table 2.

Baselines Most previous debiasing methods involve two stages: biased instance identification and debiasing models. We select several popular methods for each stage and compare their combinations with our framework.

Here, our baseline methods for biased instance identification include:

[5] Further discussion on the model inference can be found in the Appendix A.

Table 2
The basic statistics of datasets for four NLU tasks, including the ID Set and the OOD Test Set, where Val refers to the validation set.

Task	ID Set		OOD Test Set
	Train	Val	
MNLI	392K	19K	30K
SNLI	549K	9.8K	0.7K
FEVER	242K	16K	0.7K
QQP	363K	40K	8K

Table 3
Instances of four OOD test sets.

HANS	**Sentence 1**: While the actors moved the judge shouted . **Sentence 2**: The actors moved the judge . **Label**: Contradiction
Scramble	**Sentence 1**: the girl wearing a hat is less dark than the man with a beard . **Sentence 2**: the man with a beard is less dark than the girl wearing a hat . **Label**: Contradiction
PAWS	**Sentence 1**: How do I change my SBI register mobile number ? **Sentence 2**: How do I register my SBI change mobile number ? **Label**: 0
Symm.	**Sentence 1**: Down with Love is only a book . **Sentence 2**: Down with Love is a 2003 romantic comedy film . **Label**: REFUTES

- **Known-Bias** [3,11,21]. These approaches quantify the bias degree of each training instance via data statistics or researchers' insights. Then the instances with high bias degree are regarded as biased ones and used to train a bias-only model.
- **Self-debias** [12]. These approaches train a bias-only model based on partial training data to identify biased instances automatically.

Besides, we select three widely-used debiasing methods for comparison.

- **Re-weighting (RW)** [17]. This method aims to reduce the contribution of each biased instance on the overall training loss by assigning it with a scalar weight.
- **Product-of-expert (PoE)** [17]. It trains the main model in an ensemble manner with the bias-only model, which is trained in advance and uses biased features to make predictions. By doing so, the main model is encouraged to focus on unbiased features and thus becomes more robust.
- **Confidence Regularization (CR)** [11]. It trains a bias-only model and a teacher model. The output probability distribution of the latter is adjusted with that of the former. Then the re-scaled output distribution is used to enhance a main model.

Finally, we also compare our framework with the following comparisons:

- **End2End** framework [13]. In this framework, a shallow model and a main model are simultaneously but respectively trained based on the low-layer and the top-layer sentence representations, during which these two models interchangeably re-weight the importance of instances.
- **Learning from Failure (LfF)** [42]. It simultaneously trains a biased model to amplify shortcuts and a debiased model, re-weighted to focus on examples that the biased model predicts low probabilities.
- **Just Train Twice (JTT)** [43]. It involves initially training a model for a few epochs, followed by training a second model to give greater weight to examples that were incorrectly classified by the first model.
- **Minimax Training** [44]. It employs a minimax objective to train a learner and an auxiliary model simultaneously. The auxiliary model seeks to maximize the learner's loss by assigning higher weights to examples for which the learner predicts low probabilities. Meanwhile, the learner concentrates on these examples to minimize its loss.

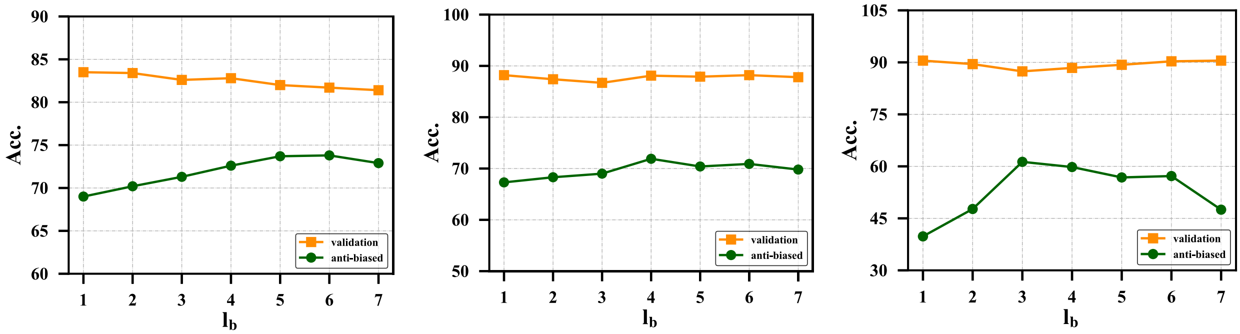
To facilitate the subsequent descriptions, we name our framework as **DeRC**. Besides, we report the performance of a variant of our framework: **DePoE**. In this variant, we first identify biased training instances according to the low-layer output probabilities and then apply the PoE method to debias the model.

The key difference between DeRC and most previous debiasing methods lies in how they handle biased instances during training. Unlike the baseline methods that rely on explicitly identifying biased training instances through techniques like data analysis or separate bias-only models, DeRC leverages the inherent difference in how low and high layers of the Transformer model learn biased

Table 4

Experimental results on the validation sets and OOD test sets: 1) the validation sets of MNLI, FEVER, QQP, SNLI; 2) their corresponding OOD test sets. The results for QQP are directly cited from [13] and the other results are cited from the corresponding papers. Values of Δ denote the performance gaps between debiasing methods and BERT on the OOD test sets. The best results are in **bold**, and the second best are underlined.

Model	MNLI			FEVER			QQP			SNLI		
	Val	HANS	Δ	Val	Symm.	Δ	Val	PAWS	Δ	Val	Scramble	Δ
BERT	**84.5**	62.3	-	85.9	64.4	-	**91.0**	33.5	-	**90.6**	72.7	-
Known-Bias + RW [17]	83.5	69.2	+6.9	84.6	66.5	+2.1	-	-	-	86.4	80.3	+7.6
Known-Bias + PoE [17]	82.9	67.9	+5.6	86.4	69.1	+4.7	-	-	-	87.2	77.4	+4.7
Known-Bias + CR [11]	**84.5**	69.1	+6.8	86.4	66.2	+1.6	89.1	40.0	+6.5	**90.6**	83.2	+10.5
Self-debias + RW [12]	82.3	69.7	+7.4	87.1	65.5	+1.1	85.2	57.4	+23.9	-	-	-
Self-debias + PoE [12]	81.9	66.8	+4.5	85.9	65.8	+1.4	-	-	-	-	-	-
Self-debias + CR [12]	84.3	67.1	+4.8	87.5	66.0	+1.6	89.0	43.0	+9.5	89.4	78.3	+5.6
End2End [13]	83.2	71.2	+8.9	86.9	-	-	90.2	46.5	+13.0	-	-	-
LfF [42]	83.4	69.1	+6.8	87.1	66.8	+2.4	89.2	47.3	+13.8	86.8	81.1	+8.4
JTT [43]	82.1	67.7	+5.4	86.9	66.1	+1.7	88.8	45.1	+11.6	87.3	80.2	+7.5
Minmax Training [44]	83.6	**72.8**	**+10.5**	85.4	68.5	+4.1	87.9	53.7	+20.2	-	-	-
DePoE	83.6	62.6	+0.3	78.0	68.0	+3.6	79.7	59.2	+25.7	81.2	81.0	+8.3
DeRC	82.8	72.6	+10.3	**88.1**	**71.9**	**+7.5**	88.4	**59.8**	**+26.3**	89.2	**88.3**	**+15.6**



(a) NLI (b) FactVer (c) ParaIden

Fig. 4. The performance of DeRC with different selected low layer l_b. Green lines denote the results of the validation sets, and orange ones denote those of the OOD test sets.

versus unbiased features. Specifically, DeRC adds a classifier on a selected low layer to encourage the low-layer representation to encode more common biased features, and then introduces a residual connection to the top-layer classifier, forcing it to focus on unbiased features. This self-debiasing approach allows DeRC to avoid the need for external bias identifiers or complex ensembling techniques used by some baselines, like the End2End framework.

Implementations details To ensure fair comparison, we use `BERT-base` to develop DeRC. During the process of fine-tuning models on each dataset, we follow the standard setup [2] to construct inputs and use the hidden state of token `[CLS]` for classification. For each task, we use a batch size of 32 and fine-tune the model for 5 epochs with the learning rate 5e-5. Besides, we select Adam [45] as the optimizer to update parameters. To select hyperparameters, we search for the optimal settings for a BERT-based NLU model across the validation sets of four tasks. Specifically, we use grid search to find the hyperparameters that lead to the best performance on these validation sets, and then apply these optimal hyperparameters across all pretrained models.

We evaluate the model performance on the validation sets and the corresponding OOD test sets. Following Utama et al. [12], we use accuracy (Acc.) as the main metric for three tasks. In addition, we evaluate the interpretability results on the QQP dataset using the metrics proposed by Wang et al. [46]. Please see Section 5.4 for details.

5.2. Effect of the chosen low layer l_b

Under our framework, the chosen low layer l_b is a crucial hyperparameter for biased sentence representation learning. Based on our analysis in Section 3.2, we argue that the performance gap of the low-layer classifier between the biased and anti-biased instances should be as large as possible. In this way, the classifier of low-layer l_b would encode sufficient biased features while encoding less task-relevant useful unbiased features. However, a too-small value of l_b is not an ideal choice, since such a low-layer classifier will not comprehensively capture biased features. As demonstrated in [17], the lowest layers, like the 1st and 2nd layers, would ignore syntactic features, which may also belong to biased features. Therefore, we need to choose a layer whose representation may contain more potentially biased features. Finally, taking the result of Fig. 2 into account as well, we select l_b as 4.

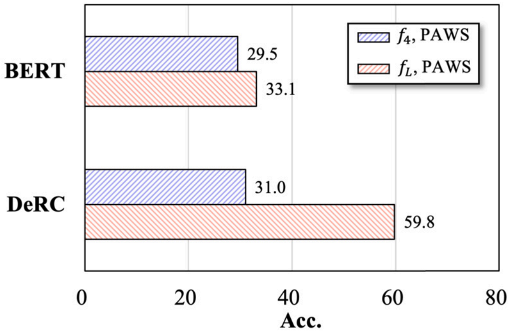


Fig. 5. Performance on PAWS of the 4th-layer and top-layer classifiers in BERT and DeRC.

To further examine the impact of l_b, we vary its value from 1 to 7 and evaluate the performance of DeRC on the validation and the OOD test sets. As shown in Fig. 4, DeRC consistently performs well on all OOD test sets for all l_b, indicating that l_b is task-agnostic and generic for all datasets. Furthermore, we find that the model's performance deteriorates slightly when l_b is within the range of 1 to 2, supporting our belief that l_b should not be too small.

5.3. Main results

As shown in Table 4,[6] DeRC achieves the best performance on most of the OOD test sets. In particular, on the Symm. set, DeRC improves accuracy by 7.5% than BERT, while the previous best model (Known-Bias+PoE) only brings a gain of 4.7%. Besides, DeRC reaches the best performance on PAWS, surpassing the previous best work (Known-Bias+CR), while bringing a much less performance drop on QQP (2.6% v.s. 5.8%). The results on Scramble also demonstrate the superiority of DeRC. In addition, DeRC achieves the most significant accuracy improvement on FEVER. Thus, we confirm that DeRC is effective in improving the model performance on OOD test sets and harmless on the validation sets.

Furthermore, the comparison between DeRC and its variant DePoE also proves the effectiveness of the residual connection. Unlike DeRC uses a residual connection, DePoE uses the probabilities output by the low-layer classifier to debias models. The results show that the residual connection enables DeRC to achieve a better trade-off between the performance drop on the validation sets and the improvement on the OOD test sets.

5.4. Analysis

Moreover, we conduct more analyses to verify the effectiveness of DeRC.

Impact of residual connection In this experiment, we use QQP to train BERT and DeRC, and compare their prediction accuracies on PAWS. Similar to DeRC, we stack two classifiers on the two layers of BERT: one is the top-layer classifier, and the other is the 4th-layer classifier. As shown in Fig. 5, the accuracy gap between the 4th-layer and top-layer classifiers of DeRC is more significant than that of BERT. Note that BERT and DeRC are similar in architecture, and the only difference is that DeRC introduces a residual connection to debias the top-layer sentence representation. Thus, we confirm that the residual connection significantly improves the model generalizability.

Interpretability evaluation In the field of post-hoc interpretation research, many studies intend to interpret the model prediction by assigning each input token with an importance score, which quantifies its impact on the prediction [47–49]. In this way, the most important tokens can form the rationale supporting the prediction. Inspired by these studies, we use QQP to train DeRC and then report interpretability results on the validation set released by Wang et al. [46], which provides annotated rationales and corresponding evaluation metrics for interpretability.

Concretely, we adopt the attention-based interpretation method [49] to assign input tokens with importance scores and then follow Wang et al. [46] to select the top-k important tokens as the rationale. Afterwards, as implemented in [50,46], we use four metrics to evaluate the model interpretability from the perspective of plausibility and faithfulness. See Table 5.

- **Token-F1**. It is used to evaluate plausibility by measuring the token overlap between the model-generated and human-annotated rationales. The higher the Token-F1 is, the more plausible the rationale is.
- **MAP**. This metric measures the consistency of rationales under perturbations, and is used to evaluate faithfulness. A high MAP represents high faithfulness.

[6] Some results from previous works are missing because they did not report. We have not reproduced these results due to the absence of sufficient detail for reproduction in their respective papers.

Table 5
Evaluation results of interpretability. The metric with ↓ means the lower the score is, the better the performance achieves. For all other metrics, a high score represents good performance.

Models	Acc.	Plausibility	Faithfulness		
		Token F1	MAP	Suff. ↓	Comp.
BERT	90.07%	58.31%	71.24%	0.1531	0.3217
DeRC	**91.13%**	**62.25%**	**75.62%**	**0.0922**	**0.3843**

Table 6
Experimental results on two sets: 1) the validation sets of MNLI, FEVER, QQP, SNLI; 2) their corresponding OOD test sets. Values of Δ denote the performance gaps between debiasing methods and RoBERTa-base on the OOD test sets.

Model	MNLI			FEVER			QQP			SNLI		
	Val	HANS	Δ	Val	Symm.	Δ	Val	PAWS	Δ	Val	Scramble	Δ
RoBERTa	**87.2**	73.5	-	**89.3**	66.3	-	**91.5**	40.1	-	**90.6**	72.7	-
w/ DePoE	84.9	75.2	+1.7	87.2	69.4	+3.1	82.7	58.5	+18.4	81.2	81.0	+8.3
w/ DeRC	86.4	**78.1**	**+4.6**	88.1	**72.9**	**+6.6**	89.2	**60.5**	**+20.4**	89.2	**88.3**	**+15.6**
DeBERTa-base	**90.2**	78.2	-	89.1	69.1	-	**92.3**	45.3	-	**91.5**	82.3	-
w/ DePoE	85.1	82.1	+3.9	86.5	71.0	+1.9	85.5	61.7	+16.4	88.7	88.4	+6.1
w/ DeRC	89.7	**84.5**	**+6.3**	**90.1**	**79.6**	**+10.5**	91.0	**67.3**	**+22.0**	90.8	**90.1**	**+7.8**
DeBERTa-large	**90.2**	79.2	-	**90.1**	72.3	-	**92.3**	49.4	-	**92.0**	90.4	-
w/ DePoE	86.9	85.4	+6.2	88.7	77.7	+5.4	86.1	67.9	+18.5	91.5	91.1	+0.7
w/ DeRC	90.2	**88.5**	**+9.3**	91.5	**84.1**	**+11.8**	91.7	**72.1**	**+22.7**	91.9	**91.7**	**+1.3**

- **Sufficiency** (Suff.) and **Comprehensiveness** (Comp.). Both two metrics are used to assess the degree of the provided rationale reflecting the prediction. A faithful rationale should have a low sufficiency score and a high comprehensiveness score.

From Table 5, we can find that DeRC outperforms BERT on all metrics, that is, the rationales provided by DeRC are more plausible and faithful. Thus, we confirm that DeRC can improve the model ability of understanding.

5.5. Results based on advanced pretrained transformer models

To verify the generalizability of DeRC across multiple pretrained models, we develop DeRC and DePoE based on three representative pretrained models: RoBERTa-base, DeBERTa-base and DeBERTav2-xxlarge. Notably, DeBERTav2-xxlarge is one of the largest transformer models available, with 1.5 billion parameters. This selection allows us to explore DeRC's generalizability in terms of both model family and scale. We re-conduct experiments using the same hyperparameters as BERT-base and experimental results are shown in Table 6. Overall, DeRC consistently achieves the best performance on all OOD test sets, which indicates the superior generalizability of DeRC.

6. Conclusions

In this work, we have proposed DeRC for Transformer-based NLU models, which utilizes the biased sentence representation learned by the low-layer classifier to debias the top-layer sentence representation. Compared with previous studies, DeRC is more efficient as it does not require manual analysis or the use of an additional bias-only model. We conduct extensive experiments on commonly-used datasets of three NLU tasks. Experimental results show that DeRC can achieve better performance on OOD test sets, while maintaining comparable performance on validation sets. In addition, DeRC can improve the ability of understanding.

In the future, we will continue to explore the low-layer representations for better performance trade-off between the validation and OOD test sets during inference. In addition, we plan to apply DeRC to other NLU tasks, such as sentiment analysis, machine reading comprehension, and so on. Finally, we will study whether DeRC is suitable for natural language generation tasks.

CRediT authorship contribution statement

Xiaoyue Wang: Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Xin Liu:** Formal analysis, Investigation, Methodology, Software, Visualization, Writing – review & editing. **Lijie Wang:** Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Suhang Wu:** Writing – review & editing. **Jinsong Su:** Project administration, Supervision, Writing – original draft, Writing – review & editing. **Hua Wu:** Supervision.

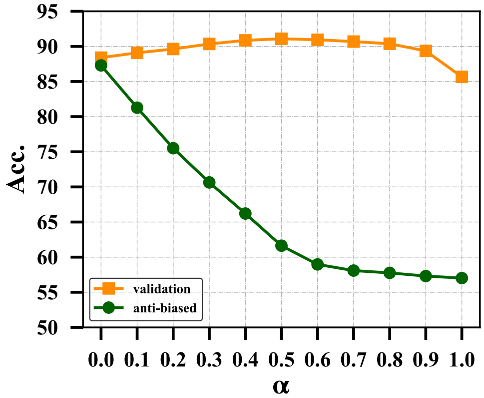


Fig. A.6. Performance of models with different α on the MNLI validation set and anti-biased validation instances.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Jinsong Su reports financial support was provided by National Natural Science Foundation of China (No. 62276219), and Natural Science Foundation of Fujian Province of China (No. 2024J01613139). If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The project was supported by National Natural Science Foundation of China (No. 62276219), and Natural Science Foundation of Fujian Province of China (No. 2024J01613139).

Appendix A. Further discussion about the model inference

We notice that in addition to the common biased features, the low-layer sentence representation also contains useful unbiased features. Thus, Equation (3) not only encourages the top-layer sentence representation to ignore the common biased features but also makes it discard some useful unbiased features, of which the amount is less than that of biased features, as analyzed in Section 3.2. To deal with this issue, we reincorporate the low-layer sentence representation into the top-layer classifier in a weighting manner:

$$p_L = \mathrm{Softmax}(W_L(\alpha * h_{\mathrm{CLS}}^{l_b} + (1-\alpha) * h_{\mathrm{CLS}}^{L}) + b_L) \tag{A.1}$$

where α is used to control the effect of $h_{\mathrm{CLS}}^{l_b}$ during inference.

Then, we vary α from 0 to 1 with an interval of 0.1, and compare the model performance on both the validation set and anti-biased validation instances. As shown in Fig. A.6, although the use of low-layer sentence representation slightly improves the model's performance on the validation set, it significantly degrades the performance on the anti-biased instances as α increases. Therefore, we directly set α to 0 in subsequent experiments. In other words, we will use the top-layer sentence representation for predictions during inference.

Appendix B. Experiments based on RoBERT-base and DeBERTa-base

Following the preliminary experiments setting in Section 3.2, we further conduct experiments to analyze the feature learning of different layers of RoBERTa-base [51] and DeBERTa-base-v3 [52] from the perspectives of model training and prediction. From Fig. B.7, Fig. B.9, Fig. B.8 and Fig. B.10, we can find that the training losses and prediction accuracies of both the RoBERTa DeBERTa exhibit almost the same trends as those of BERT.

Data availability

Data will be made available on request.

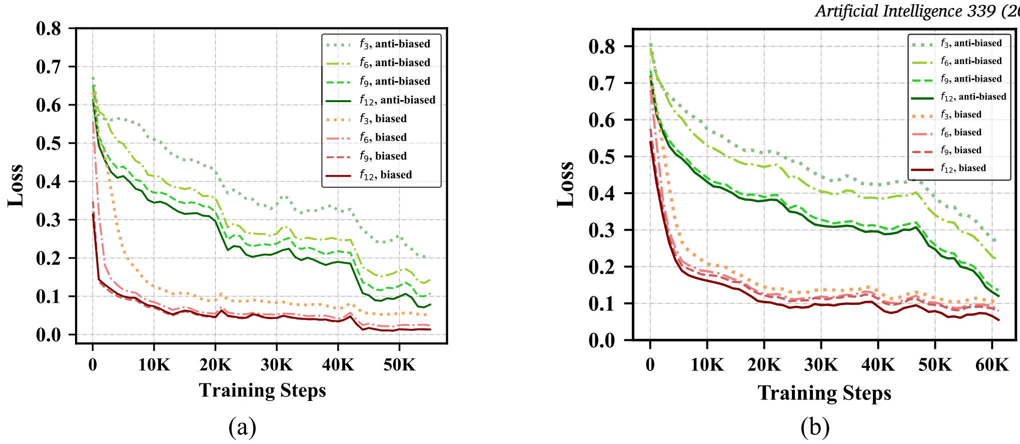


Fig. B.7. RoBERTa training loss curves on biased and anti-biased training instances of QQP (a) and MNLI (b), where f_i represents the i-th layer classifier.

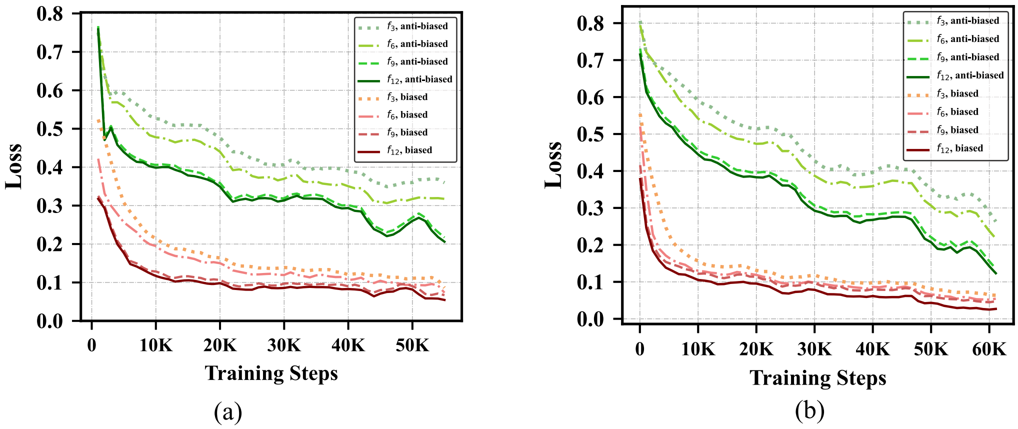


Fig. B.8. DeBERTa training loss curves on biased and anti-biased training instances of QQP (a) and MNLI (b), where f_i represents the i-th layer classifier.

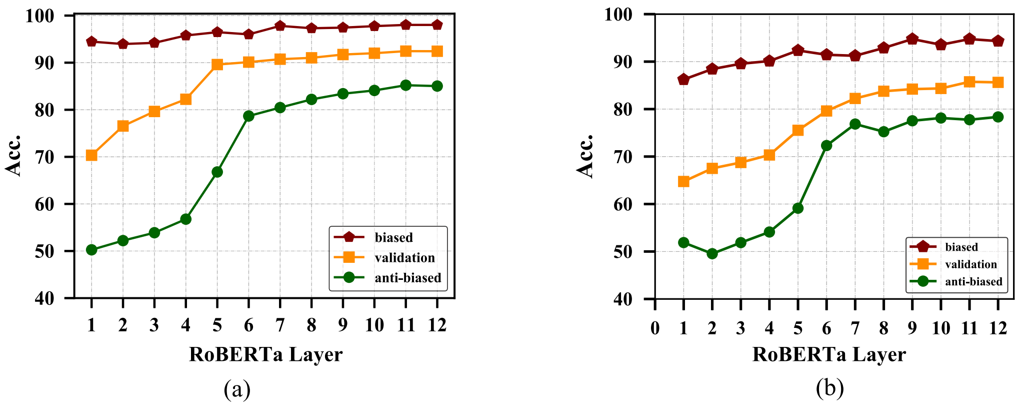


Fig. B.9. The prediction performance of layer-specific classifiers from RoBERTa-based NLU models on the validation set, the biased validation instances, and the anti-biased validation instances of QQP (a) and MNLI (b).

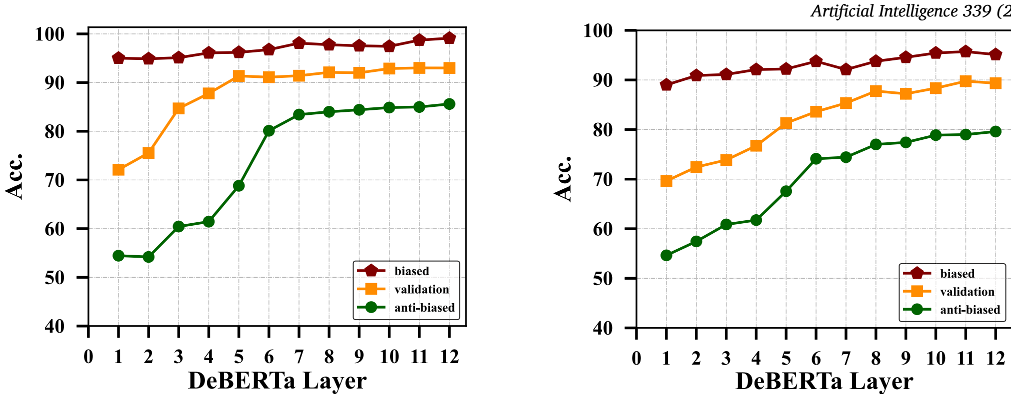


Fig. B.10. The prediction performance of layer-specific classifiers from DeBERTa-based NLU models on the validation set, the biased validation instances, and the anti-biased validation instances of QQP (a) and MNLI (b).

References

[1] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, S. Bowman, GLUE: a multi-task benchmark and analysis platform for natural language understanding, in: Workshop of EMNLP 2018, 2018.
[2] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: NAACL 2019, 2019.
[3] S. Gururangan, S. Swayamdipta, O. Levy, R. Schwartz, S. Bowman, N.A. Smith, Annotation artifacts in natural language inference data, in: NAACL 2018, 2018.
[4] T. McCoy, E. Pavlick, T. Linzen, Right for the wrong reasons: diagnosing syntactic heuristics in natural language inference, in: ACL 2019, 2019.
[5] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, F.A. Wichmann, Shortcut learning in deep neural networks, Nat. Mach. Intell. (2020).
[6] R. Zellers, Y. Bisk, R. Schwartz, Y. Choi, SWAG: a large-scale adversarial dataset for grounded commonsense inference, in: EMNLP 2018, 2018.
[7] Y. Zhang, J. Baldridge, L. He, PAWS: paraphrase adversaries from word scrambling, in: NAACL 2019, 2019.
[8] A. Poliak, J. Naradowsky, A. Haldar, R. Rudinger, B. Van Durme, Hypothesis only baselines in natural language inference, in: SEM 2018, 2018.
[9] M. Tsuchiya, Performance impact caused by hidden bias of training data for recognizing textual entailment, in: LREC 2018, 2018.
[10] T. Schuster, D. Shah, Y.J.S. Yeo, D. Roberto Filizzola Ortiz, E. Santus, R. Barzilay, Towards debiasing fact verification models, in: EMNLP 2019, 2019.
[11] P.A. Utama, N.S. Moosavi, I. Gurevych, Mind the trade-off: debiasing NLU models without degrading the in-distribution performance, in: ACL 2020, 2020.
[12] P.A. Utama, N.S. Moosavi, I. Gurevych, Towards debiasing NLU models from unknown biases, in: EMNLP 2020, 2020.
[13] A. Ghaddar, P. Langlais, M. Rezagholizadeh, A. Rashid, End-to-end self-debiasing framework for robust NLU training, in: Findings of ACL 2021, 2021.
[14] L. Du, X. Ding, Z. Sun, T. Liu, B. Qin, J. Liu, Towards stable natural language understanding via information entropy guided debiasing, in: A. Rogers, J.L. Boyd-Graber, N. Okazaki (Eds.), Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023, Association for Computational Linguistics, 2023, pp. 2868–2882.
[15] W. Xu, Q. Liu, S. Wu, L. Wang, Counterfactual debiasing for fact verification, in: A. Rogers, J.L. Boyd-Graber, N. Okazaki (Eds.), Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023, Association for Computational Linguistics, 2023, pp. 6777–6789.
[16] R. Karimi Mahabadi, Y. Belinkov, J. Henderson, End-to-end bias mitigation by modelling biases in corpora, in: ACL 2020, 2020.
[17] C. Clark, M. Yatskar, L. Zettlemoyer, Don't take the easy way out: ensemble based methods for avoiding known dataset biases, in: EMNLP 2019, 2019.
[18] H. He, S. Zha, H. Wang, Unlearn dataset bias in natural language inference by fitting the residual, in: Workshop of DeepLo 2019, 2019.
[19] G. Jawahar, B. Sagot, D. Seddah, What does bert learn about the structure of language?, in: ACL 2019, 2019.
[20] I. Tenney, D. Das, E. Pavlick, BERT rediscovers the classical NLP pipeline, in: ACL 2019, 2019.
[21] M. Du, V. Manjunatha, R. Jain, R. Deshpande, F. Dernoncourt, J. Gu, T. Sun, X. Hu, Towards interpreting and mitigating shortcut learning behavior of NLU models, in: NAACL 2021, 2021.
[22] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: CVPR 2016, 2016.
[23] R. Snow, L. Vanderwende, A. Menezes, Effectively using syntax for recognizing false entailment, in: NAACL 2006, 2006.
[24] R. Zellers, A. Holtzman, Y. Bisk, A. Farhadi, Y. Choi, HellaSwag: can a machine really finish your sentence?, in: ACL 2019, 2019.
[25] C. Clark, M. Yatskar, L. Zettlemoyer, Learning to model and ignore dataset bias with mixed capacity ensembles, in: Findings of EMNLP 2020, 2020.
[26] V. Sanh, T. Wolf, Y. Belinkov, A.M. Rush, Learning from others' mistakes: avoiding dataset biases without modeling them, arXiv preprint, arXiv:2012.01300, 2020.
[27] Y. Nie, A. Williams, E. Dinan, M. Bansal, J. Weston, D. Kiela, Adversarial nli: a new benchmark for natural language understanding, in: ACL 2020, 2020.
[28] M. Lee, S. Won, J. Kim, H. Lee, C. Park, K. Jung, Crossaug: a contrastive data augmentation method for debiasing fact verification models, Inf. Knowl. Manag. (2021).
[29] X. Wang, X. Liu, L. Wang, Y. Wang, J. Su, H. Wu, IBADR: an iterative bias-aware dataset refinement framework for debiasing NLU models, in: H. Bouamor, J. Pino, K. Bali (Eds.), Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023, Association for Computational Linguistics, 2023, pp. 9176–9186.
[30] R. Sharma, J. Allen, O. Bakhshandeh, N. Mostafazadeh, Tackling the story ending biases in the story cloze test, in: ACL 2018, 2018.
[31] Y. Belinkov, A. Poliak, S. Shieber, B. Van Durme, A. Rush, On adversarial removal of hypothesis-only bias in natural language inference, in: SEM 2019, 2019.
[32] J. Stacey, P. Minervini, H. Dubossarsky, S. Riedel, T. Rocktäschel, Avoiding the hypothesis-only bias in natural language inference via ensemble adversarial training, in: EMNLP 2020, 2020.
[33] Y. Ganin, V. Lempitsky, Unsupervised domain adaptation by backpropagation, in: ICML 2015, 2015.
[34] L. Tu, G. Lalwani, S. Gella, H. He, An empirical study on robustness to spurious correlations using pre-trained language models, Trans. Assoc. Comput. Linguist. (2020).
[35] Y. Yaghoobzadeh, S. Mehri, R.T. des Combes, T.J. Hazen, A. Sordoni, Increasing robustness to spurious correlations using forgettable examples, in: ACL 2021, 2021.
[36] Y. Lyu, P. Li, Y. Yang, M. de Rijke, P. Ren, Y. Zhao, D. Yin, Z. Ren, Feature-level debiased natural language understanding, in: AAAI 2023, 2023.
[37] J. Hewitt, C.D. Manning, A structural probe for finding syntax in word representations, in: NAACL 2019, 2019.

[38] A. Williams, N. Nangia, S.R. Bowman, A broad-coverage challenge corpus for sentence understanding through inference, in: M.A. Walker, H. Ji, A. Stent (Eds.), Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers), Association for Computational Linguistics, 2018, pp. 1112–1122.

[39] S.R. Bowman, G. Angeli, C. Potts, C.D. Manning, A large annotated corpus for learning natural language inference, in: L. Màrquez, C. Callison-Burch, J. Su, D. Pighin, Y. Marton (Eds.), Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015, The Association for Computational Linguistics, 2015, pp. 632–642.

[40] I. Dasgupta, D. Guo, A. Stuhlmüller, S. Gershman, N.D. Goodman, Evaluating compositionality in sentence embeddings, in: C. Kalish, M.A. Rau, X.J. Zhu, T.T. Rogers (Eds.), Proceedings of the 40th Annual Meeting of the Cognitive, CogSci 2018, Madison, WI, USA, July 25-28, 2018, cognitivesciencesociety.org, Science Society, 2018, https://mindmodeling.org/cogsci2018/papers/0307/index.html.

[41] J. Thorne, A. Vlachos, O. Cocarascu, C. Christodoulopoulos, A. Mittal, The fact extraction and verification (fever) shared task, in: EMNLP 2018, 2018.

[42] J.H. Nam, H. Cha, S. Ahn, J. Lee, J. Shin, Learning from failure: Training debiased classifier from biased classifier, CoRR, arXiv:2007.02561, 2020, arXiv: 2007.02561.

[43] E.Z. Liu, B. Haghgoo, A.S. Chen, A. Raghunathan, P.W. Koh, S. Sagawa, P. Liang, C. Finn, Just train twice: improving group robustness without training group information, in: M. Meila, T. Zhang (Eds.), Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event, volume 139 of Proceedings of Machine Learning Research, PMLR, 2021, pp. 6781–6792, http://proceedings.mlr.press/v139/liu21f.html.

[44] M. Korakakis, A. Vlachos, Improving the robustness of NLI models with minimax training, in: A. Rogers, J.L. Boyd-Graber, N. Okazaki (Eds.), Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023, Association for Computational Linguistics, 2023, pp. 14322–14339.

[45] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, in: ICLR 2015, 2015.

[46] L. Wang, Y. Shen, S. Peng, S. Zhang, X. Xiao, H. Liu, H. Tang, Y. Chen, H. Wu, H. Wang, A fine-grained interpretability evaluation benchmark for neural nlp, arXiv preprint, arXiv:2205.11097, 2022.

[47] K. Simonyan, A. Vedaldi, A. Zisserman, Deep inside convolutional networks: visualising image classification models and saliency maps, in: ICLR 2014, 2014.

[48] D. Smilkov, N. Thorat, B. Kim, F. Viégas, M. Wattenberg, Smoothgrad: removing noise by adding noise, arXiv preprint, arXiv:1706.03825, 2017.

[49] S. Jain, B.C. Wallace, Attention is not explanation, in: NAACL 2019, 2019.

[50] J. DeYoung, S. Jain, N.F. Rajani, E. Lehman, C. Xiong, R. Socher, B.C. Wallace, ERASER: a benchmark to evaluate rationalized NLP models, in: ACL 2020, 2020.

[51] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: a robustly optimized bert pretraining approach, arXiv preprint, arXiv:1907.11692, 2019.

[52] P. He, J. Gao, W. Chen, Debertav3: improving deberta using electra-style pre-training with gradient-disentangled embedding sharing, in: The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023, OpenReview.net, 2023, https://openreview.net/pdf?id=sE7-XhLxHA.