SIMPLE BUT EFFECTIVE TECHNIQUES TO REDUCE DATASET BIASES

Anonymous authors
Paper under double-blind review

ABSTRACT

There have been several studies recently showing that strong natural language understanding (NLU) models are prone to relying on unwanted dataset *biases* without learning the underlying task, resulting in models which fail to generalize to out-of-domain datasets, and are likely to perform poorly in real-world scenarios. We propose several learning strategies to train neural models which are more robust to such biases and transfer better to out-of-domain datasets. We introduce an additional lightweight *bias-only model* which learns dataset biases and uses its prediction to adjust the loss of the base model to reduce the biases. In other words, our methods down-weight the importance of the biased examples, and focus training on *hard* examples, i.e. examples that cannot be correctly classified by only relying on biases. Our approaches are model agnostic and simple to implement. We experiment on large-scale natural language inference and fact verification datasets and their out-of-domain datasets and show that our debiased models significantly improve the robustness in all settings, including gaining 9.76 points on the FEVER symmetric evaluation dataset, 5.45 on the HANS dataset and 4.78 points on the SNLI hard set. These datasets are specifically designed to assess the robustness of models in the out-of-domain setting where typical biases in the training data do not exist in the evaluation set.

1 INTRODUCTION

Recent neural models (Devlin et al., 2019; Radford et al., 2018; Chen et al., 2017) have achieved high and even near human-performance on several large-scale natural language understanding benchmarks. However, it has been demonstrated that neural models tend to rely on existing idiosyncratic biases in the datasets, and leverage superficial correlations between the label and existing shortcuts in the training dataset to perform surprisingly well[1], without learning the underlying task (Kaushik & Lipton, 2018; Gururangan et al., 2018; Poliak et al., 2018; Schuster et al., 2019; Niven & Kao, 2019; McCoy et al., 2019). For instance, natural language inference (NLI) consists of determining whether a hypothesis sentence (*There is no teacher in the room*) can be inferred from a premise sentence (*Kids work at computers with a teacher's help*)[2](Dagan et al., 2006). However, recent work has demonstrated that large-scale NLI benchmarks contain annotation artifacts; certain words in the hypothesis are highly indicative of inference class that allow models with poor premise grounding to perform unexpectedly well (Poliak et al., 2018; Gururangan et al., 2018). As an example, in some NLI benchmarks, negation words such as "nobody", "no", and "not" in the hypothesis are often highly correlated with the contradiction label. As a consequence, NLI models do not need to learn the true relationship between the premise and hypothesis and instead can rely on statistical cues, such as learning to link negation words with the contradiction label.

As a result of the existence of such biases, models exploiting statistical shortcuts during training often perform poorly on out-of-domain datasets, especially if they are carefully designed to limit the spurious cues. To allow proper evaluation, recent studies have tried to create new evaluation datasets that do not contain such biases (Gururangan et al., 2018; Schuster et al., 2019). Unfortunately, it is hard to avoid spurious statistical cues in the construction of large-scale benchmarks, and collecting

---

[1]We use biases, heuristic patterns or shortcuts interchangeably.

[2]The given sentences are in the contradictory relation and the hypothesis cannot be inferred from the premise.

new datasets is costly (Sharma et al., 2018). It is therefore crucial to develop techniques to reduce the reliance on biases during the training of the neural models.

In this paper, we propose several end-to-end debiasing techniques to adjust the cross-entropy loss to reduce the biases learned from datasets, which work by down-weighting the biased examples so that the model focuses on learning hard examples. Figure 1 illustrates an example of applying our strategy to prevent an NLI model from predicting the labels using existing biases in the hypothesis. Our strategy involves adding a bias-only branch f_B on top of the base model f_M during training (In case of NLI, the bias-only model only uses the hypothesis). We then compute the combination of the two models f_C in a way to motivate the base model to learn different strategies than the ones used by the bias-only branch f_B. At the end of the training, we remove the bias-only classifier and use the predictions of the base model.

We propose three main debiasing strategies, detailed in Section 2.2. In our first two proposed methods, the combination is done with an ensemble method which combines the predictions of the base and the bias-only models. The training loss of the base model is then computed on the output of this combined model f_C. This has the effect of reducing the loss going from the combined model to the base model for the examples which the bias-only model classifies correctly. For the third method, the bias-only predictions are used to directly weight the loss of the base model, explicitly modulating the loss depending on the accuracy of the bias-only model. All strategies work by allowing the base model to focus on learning the hard examples, by preventing it from learning the biased examples.
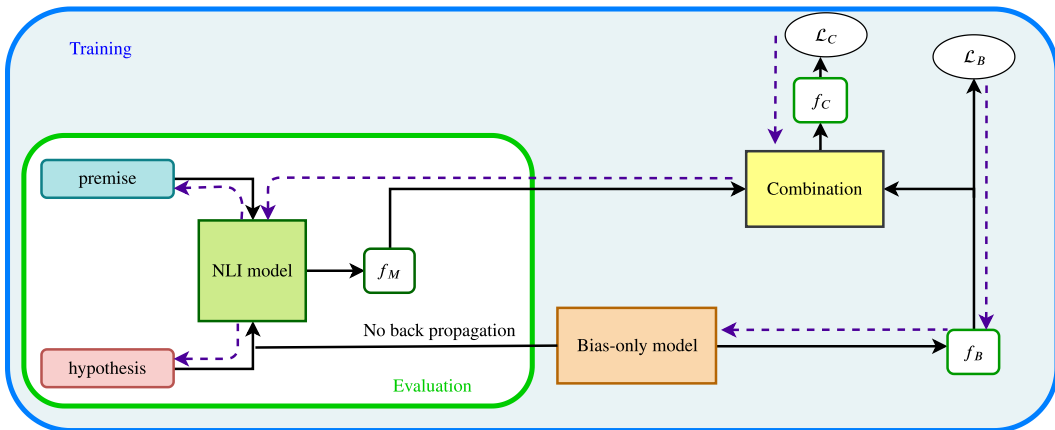

Figure 1: An illustration of our debiasing strategies on NLI. Solid arrows show the flow of input information, and dotted arrows show the back-propagation flow of error. The combination is done in different ways for different models. Blue highlighted modules are removed after training. At test time, only the predictions of the base model f_M are used.

Our approaches are simple and highly effective. They require training a simple classifier on top of the base model. Furthermore, our methods are model agnostic and general enough to be applicable for addressing common biases seen in several datasets in different domains.

We evaluate our models on challenging benchmarks in textual entailment and fact verification. For entailment, we run extensive experiments on HANS (Heuristic Analysis for NLI Systems) (McCoy et al., 2019), and hard NLI sets of Stanford Natural Language Inference (SNLI) (Bowman et al., 2015) and MultiNLI (MNLI) (Williams et al., 2018) datasets (Gururangan et al., 2018). We additionally construct hard MNLI datasets from MNLI development sets to facilitate the out-of-domain evaluation on this dataset[3]. Furthermore, we evaluate our fact verification models on *FEVER Symmetric test set* (Schuster et al., 2019). The selected datasets are highly challenging and have been carefully designed to be unbiased to allow proper evaluation of the out-of-domain performance of the models. We show that including our strategies on training baseline models including BERT (Devlin et al., 2019) provide substantial gain on out-of-domain performance in all the experiments.

---

[3]Removing the need of submitting to an online evaluation system for MNLI hard test sets.

In summary, we make the following contributions: 1) Proposing several debiasing strategies to train neural models that make them more robust to existing biases in the dataset. 2) An empirical evaluation of the proposed methods on two large-scale NLI benchmarks and obtaining substantial gain on their challenging out-of-domain data, including 5.45 points on HANS and 4.78 points on SNLI hard set. 3) Evaluating our models on fact verification, obtaining 9.76 points gain on FEVER symmetric test set, improving the results of prior work by 4.65 points.

To facilitate future work, we release our datasets and code.

2   REDUCING BIASES

Problem formulation We consider a general multi-class classification problem. Given a dataset $\mathcal{D} = \{x_i, y_i\}_{i=1}^{N}$ consisting of the input data $x_i \in \mathcal{X}$, and labels $y_i \in \mathcal{Y}$, the goal of the base model is to learn a mapping f_M parameterized by θ_M which computes the predictions over the label space given the input data, shown as $f_M : \mathcal{X} \to \mathbb{R}^{|\mathcal{Y}|}$. Our goal is to optimize θ_M parameters such that we build a model which is more resistant to benchmark biases to improve its robustness to domain changes when the typical biases observed in the training data do not exist in the evaluation dataset.

The key idea of our approach, depicted in Figure 1 is first to identify the dataset biases and heuristic patterns which the base model is susceptible to relying on. Then, we use a bias-only branch to capture these biases. We propose several strategies to incorporate the bias-only knowledge into the training of the base model to make a robust version of it. After training we remove the bias-only model and use the predictions of the base model. In this section, we explain each of these components.

2.1   BIAS-ONLY BRANCH

We assume that we do not have access to any data from the out-of-domain dataset, so we need to know a priori about the possible types of shortcut patterns we would like the base model to avoid relying on them. Once these shortcut patterns are identified, we train a bias-only model designed to capture the identified biases which only uses the *biased features*. For instance, it has been shown that a hypothesis-only model in the large-scale NLI datasets can correctly classify the majority of samples using the artifacts (Poliak et al., 2018; Gururangan et al., 2018). Therefore, our bias-only model for NLI only uses hypothesis sentences. But note that the bias-only model can, in general, have any form, and is not limited to models which are using only a part of input data. Let $x_i^b \in \mathcal{X}^b$ be *biased features* of x_i which are predictive of y_i. We then formalize this bias-only model as a mapping $f_B : \mathcal{X}^b \to \mathbb{R}^{|\mathcal{Y}|}$ parameterized by θ_B trained using cross-entropy loss $\mathcal{L}_B$:

$$\mathcal{L}_B(\theta_B) = -\frac{1}{N} \sum_{i=1}^{N} \boldsymbol{a}_i \log(softmax(f_B(x_i^b))), \tag{1}$$

where $\boldsymbol{a}_i$ is the one-hot representation of the true label for the i$^{\text{th}}$ example. In the next section, we explain how we use the bias-only model to make a robust version of the base model.

2.2   PROPOSED DEBIASING STRATEGIES

We propose several strategies to incorporate the bias-only f_B knowledge into training of the base model f_M and update its parameters θ_M using the obtained loss $\mathcal{L}_C$ of the combined classifier f_C. All these strategies have the form illustrated in Figure 1, where the predictions of the bias-only model are combined with either the predictions of the base model or its error to down-weight the loss from the biased examples, thereby affecting the error backpropagated into the base model.

As also illustrated in Figure 1, it is often convenient for the bias-only model to share parameters with the base model, such as sharing a sentence encoder. To prevent the base model from learning the biases, the bias-only loss $\mathcal{L}_B$ is not back-propagated to these shared parameters of the base model. To accommodate this sharing, the bias-only and the base models are trained together. Next, we explain how the loss of the combined classifier, $\mathcal{L}_C$, is computed for each of our debiasing methods.

2.2.1 METHOD 1: PRODUCT OF EXPERTS

Our first approach is based on the idea of the product of experts ensemble method (Hinton, 2002): "It is possible to combine multiple probabilistic models of the same data by multiplying the probabilities together and then renormalizing.". Here, we use this notion to combine the bias-only and base model predictions by computing the element-wise product $\odot$ between their predictions as $f_B(x_i^b) \odot f_M(x_i)$. We compute this combination in the logarithmic space, which works better in practice:

$$f_C(x_i, x_i^b) = \log(softmax(f_B(x_i^b))) + \log(softmax(f_M(x_i))), \tag{2}$$

The key intuition behind this model is to combine the probability distributions of the bias-only and the base model to allow them to make predictions based on different characteristics of the input; the bias-only branch covers prediction based on biases, and the base model focuses learning the actual task. We then compute $\mathcal{L}_C$ as the cross-entropy loss of the combined predictions f_C. Then the base model parameters θ_M are trained using the cross-entropy loss of the combined classifier f_C:

$$\mathcal{L}_C(\theta_M; \theta_B) = -\frac{1}{N} \sum_{i=1}^{N} \boldsymbol{a}_i \log(softmax(f_C(x_i, x_i^b))), \tag{3}$$

When this loss is backpropagated to base model parameters θ_M, the predictions of the bias-only model decrease the updates for examples which it can accurately predict.

2.2.2 METHOD 2: RUBI VARIATIONS (CADENE ET AL., 2019)

Recently, Cadene et al. (2019) propose a model called *RUBI* to alleviate unimodal biases learned by Visual Question Answering (VQA) models. Cadene et al. (2019)'s study is limited to alleviating biases in VQA benchmarks. We, however, evaluate the effectiveness of their formulation together with our newly proposed variations in the natural language understanding context on several challenging NLU datasets.

We first apply a sigmoid function to the bias-only model's predictions to obtain a mask containing an importance weight between 0 and 1 for each possible label. We then compute the element-wise product between the obtained mask and the base model's predictions:

$$f_C(x_i, x_i^b) = f_M(x_i) \odot \sigma(f_B(x_i^b)), \tag{4}$$

The main intuition is to dynamically adjust the predictions of the base model to prevent the base model from leveraging the shortcuts. We note two properties of this loss. (1) When the bias-only model correctly classifies the example, the mask increases the value of the correct prediction while decreases the scores for other labels. As a result, the loss of biased examples is down-weighted. (2) For the hard examples that cannot be correctly classified using bias-only model, the obtained mask increases the score of the wrong answer. This, in turn, increases the contribution of hard examples and encourages the base model to learn the importance of correcting them. We additionally propose the following new variants of this model:

1. Computing the combination in logarithmic space, which we refer to it as *RUBI + log space*.

$$f_C(x_i, x_i^b) = \log(softmax(f_M(x_i))) + \log(\sigma(softmax(f_B(x_i^b)))). \tag{5}$$

2. Normalizing the output of the bias-only model, followed by RUBI model, which we refer to it as *RUBI + normalize*:

$$f_C(x_i, x_i^b) = f_M(x_i) \odot \sigma(softmax(f_B(x_i^b))) \tag{6}$$

As with our first method, we then update the parameters of the base model θ_M by backpropagating the cross-entropy loss $\mathcal{L}_C$ of the combined classifier.

2.2.3 METHOD 3: DEBIASED FOCAL LOSS

Focal loss was originally proposed in Lin et al. (2017) to improve a single classifier by down-weighting the well-classified points. We propose a novel variant of this loss, in which we leverage

the bias-only branch's predictions to reduce the relative importance of the most biased examples and allow the model to focus on learning the *hard* examples. We define *Debiased Focal Loss* as:

$$\mathcal{L}_C(\theta_M; \theta_B) = -\frac{1}{N} \sum_{i=1}^{N} \boldsymbol{a}_i (1 - softmax(f_B(x_i^b)))^\gamma \log(softmax(f_M(x_i))), \tag{7}$$

where γ is the focusing parameter, which impacts the down-weighting rate. When γ is set to 0, our Debiased Focal Loss is equivalent to the normal cross-entropy loss. For $\gamma > 0$, as the value of γ is increased, the effect of down-weighting is increased. We set $\gamma = 2$ through all experiments, which works well in practice and avoid fine-tuning it further. We note the properties of the Debiased Focal Loss: (1) When the example x_i is unbiased, and bias-only branch does not do well, $softmax(f_B(x_i^b))$ is small, therefore the scaling factor is close to 1, and the loss remains unaffected. (2) As the sample is more biased and $softmax(f_B(x_i^b))$ is closer to 1, the modulating factor approaches 0 and the loss for the most biased examples is down-weighted.

For this debiasing strategy, Debiased Focal Loss is then used to update the parameters of the base model θ_M. Note that this loss has a different form from that used for the first two methods.

3 EXPERIMENTAL RESULTS

We provide experiments on two large-scale NLI datasets, namely SNLI and MNLI, and FEVER dataset for our fact verification experiment and evaluate the models' performance on their challenging unbiased evaluation datasets proposed very recently. In most of our experiments, we consider BERT[4] as our baseline which is known to work well for these tasks, and additionally, we have included other baselines used in the prior work to compare against them. In all the experiments, we kept the hyperparameters of baselines as the default. We include low-level details in the appendix.

3.1 FACT VERIFICATION

Dataset: FEVER dataset contains claim-evidence pairs generated from Wikipedia. Schuster et al. (2019) collect a new evaluation set for FEVER dataset to avoid the idiosyncrasies observed in the claims of this benchmark. They make the original claim-evidence pairs of FEVER evaluation dataset symmetric, by augmenting the dataset and making each claim and evidence appear with each label. Therefore, by balancing the artifacts, relying on cues from claim to classify samples is equivalent to a random guess. The collected dataset is challenging and the performance of the models evaluated on this dataset drop significantly.

Base models: We consider BERT as the baseline, which works the best on this dataset (Schuster et al., 2019), and predicts the relations based on the concatenation of the claim and the evidence with a delimiter token (see Appendix A).

Bias-only model: The bias-only model predicts the labels using only claims as input.

Results: Table 1 shows the results. The obtained improvement of our debiasing methods varies between 1.11-9.76 absolute points. The Product of experts and Debiased Focal loss are highly effective, boosting the performance of the baseline model by 9.76 and 7.53 absolute points respectively, significantly surpassing the prior work (Schuster et al., 2019).

3.2 TEXTUAL ENTAILMENT

Datasets: We evaluate on hard SNLI and MNLI datasets (Gururangan et al., 2018) which are the split of these datasets where a hypothesis-only model cannot correctly predict the labels. Gururangan et al. (2018) show that the success of the recent textual entailment models is attributed to the *biased* examples, and the performance of these models are substantially lower on *hard* sets.

Base models: We consider InferSent (Conneau et al., 2017), and BERT as our base models. We choose InferSent to be able to compare against the prior work (Belinkov et al., 2019b).

Bias-only model: The bias-only model only uses the hypothesis to predict the labels (see Appendix B).

---

[4]https://github.com/huggingface/pytorch-pretrained-BERT

Table 1: Results on FEVER development (Dev) set and FEVER symmetric test set.

Debiasing method	Dev	Symmetric test set
None	85.99	56.49
RUBI	86.23	57.60
RUBI + log space	86.59	59.27
RUBI + normalize	86.16	60.11
Debiased Focal Loss	83.07	64.02
Product of experts	86.46	66.25
Schuster et al. (2019)	84.6	61.6

Table 2: Results on SNLI and SNLI hard sets.

| Debiasing method | BERT | | InferSent | |
	Test	Hard	Test	Hard
None	90.53	80.53	84.24	68.91
RUBI + log space	90.74	81.32	83.67	69.0
RUBI	90.69	80.62	83.93	69.64
RUBI + normalize	90.70	80.83	83.6	69.24
Debiased Focal Loss	89.57	83.01	73.54	73.05
Product of experts	90.11	82.15	80.35	73.69
AdvCls (Belinkov et al., 2019b)	-	-	83.56	66.27
AdvDat (Belinkov et al., 2019b)	-	-	78.30	55.60

Results on SNLI: Table 2 shows the results on SNLI dataset. For InferSent model, Debiased Focal Loss and Product of experts methods result in 4.14 and 4.78 points gain. Similarly, for the BERT model, Debiased Focal loss and Product of experts improve the results the most by 2.48 and 1.62 absolute points. Comparing to the results of Belinkov et al. (2019b), our product of expert model obtains a 7.42 point gain, significantly surpassing the prior work.

Results on MNLI: We construct hard sets from MNLI development set for both MNLI Matched and MNLI Mismatched datasets. Following Gururangan et al. (2018), we train a `fastText` classifier (Joulin et al., 2017), to predict the labels using only the hypothesis and consider the subset of the samples on which our trained hypothesis-only classifier failed as hard examples. Table 3 shows the results on the development sets and their corresponding hard sets. For BERT baseline, on MNLI matched hard dataset, the product of experts and RUBI+normalize improve the results the most by 1.46 and 1.11 points. On MNLI mismatched hard, the Debiased Focal Loss and product of experts obtain 1.37, and 1.68 points gain respectively. For InferSent baseline, on MNLI matched hard, the product of experts and RUBI improve the results by 2.34 and 0.94 points. On MNLI mismatched hard, the Product of experts and Debiased Focal Loss improve the results by 2.61 and 2.52 points.

To comply with limited access to the submission system of MNLI, we evaluate only the best result of baseline and our models on the test sets. Table 4 shows the results on the MNLI test and hard sets. Our product of expert model improves the performance on MNLI matched hard set by 0.93 points and 1.08 points on MNLI Mismatched hard set while maintaining the in-domain accuracy.

Table 3: Results on MNLI matched (MNLI) and mismatched (MNLI-M) dev and their hard sets.

| Debiasing method | BERT | | | | InferSent | | | |
| | MNLI | | MNLI-M | | MNLI | | MNLI-M | |
	Dev	Hard	Dev	Hard	Dev	Hard	Dev	Hard
None	84.41	76.56	84.53	77.55	69.97	57.03	69.99	56.53
RUBI + log space	84.46	76.80	84.86	78.04	69.70	56.57	69.95	56.56
RUBI	84.48	77.13	85.17	78.63	70.51	57.97	70.53	58.08
RUBI + normalize	84.80	77.67	84.77	78.54	70.16	57.53	70.09	57.62
Debiased Focal Loss	83.72	77.37	84.85	78.92	60.78	57.88	61.12	59.05
Product of experts	84.58	78.02	84.85	79.23	66.02	59.37	65.85	59.14

Table 4: Results on MNLI matched (MNLI) and mismatched (MNLI-M) test and hard sets.

Debiasing Method	MNLI		MNLI-M	
	Test	Hard	Test	Hard
None	84.11	75.88	83.51	75.75
Product of experts	84.11	76.81	83.47	76.83

Table 5: Results on MNLI matched dev and HANS datasets including its individual heuristic subsets.

Debiasing Method	MNLI	HANS	Constituent	Lexical	Subsequence
None	83.99	61.10	61.11	68.97	53.21
RUBI + log space	84.56	62.53	57.77	75.18	54.65
RUBI	83.93	60.35	56.51	71.09	53.44
RUBI + normalize	84.15	59.71	55.74	70.01	53.37
Debiased Focal Loss	84.33	64.99	62.42	74.45	58.11
Product of experts	84.04	66.55	64.29	77.61	57.75

3.3 SYNTACTIC BIAS

Dataset: McCoy et al. (2019) show that NLI models can rely on superficial syntactic heuristics to perform the task. They introduce HANS dataset, which covers several examples on which the models employing the syntactic heuristics fail.

Base model: We use BERT as our base model and train it on MNLI dataset.

Bias-only model: We consider several features for the bias-only model. The first three features are based on the syntactic heuristics proposed in McCoy et al. (2019): 1) Whether all the words in the hypothesis are included in the premise. 2) If the hypothesis is the contiguous subsequence of the premise. 3) If the hypothesis is a subtree in the premise's parse tree 4) The number of tokens shared between premise and hypothesis normalized by the number of tokens in the premise. We additionally include some similarity features: 5) The cosine similarity between premise and hypothesis tokens followed by mean and max-pooling. We consider the same weight for contradiction and neutral labels in the bias-only loss to allow the model to recognize entailment from not-entailment. During the evaluation, we map the neutral and contradiction labels to not-entailment.

Results: As shown in Table 5, the Product of experts and Debiased Focal loss improve the results the most by 5.45, 3.89 points. We provide the accuracy for each label on HANS dataset in Appendix C.

4 DISCUSSION

Analysis of Debiased Focal Loss To understand the impact of γ in Debiased Focal Loss, we train InferSent models with this loss for different values of γ on SNLI dataset and evaluate its performance on SNLI and SNLI hard sets. As illustrated in Figure 2, increasing γ focuses the loss on learning hard examples, and reduces the attention on learning biased examples. Consequently, the in-domain accuracy on SNLI is dropped but out-of-domain accuracy on SNLI hard set is increased.

Results: Through extensive experiments on different datasets, our methods improve out-of-domain performance in all settings. Debiased Focal Loss and Product of experts models consistently obtain the highest gains. Within RUBI variations, RUBI+log space outperforms the other variations on SNLI with BERT baseline and HANS dataset. RUBI+normalize does better than the rest on FEVER experiment and MNLI matched hard set with BERT baseline. RUBI performs the best on SNLI and MNLI experiments with InferSent baseline, and MNLI mismatched hard with BERT baseline.

As expected, improving the out-of-domain performance could come at the expense of the decreased in-domain performance, since the removed biases are useful for performing the in-domain task. This especially happens for Debiased Focal Loss, in which there is a trade-off between in-domain and out-of-domain performance as discussed depending on the parameter γ, and when the baseline model is not very powerful like InferSent. Our other models with BERT baseline consistently remain the in-domain performance.
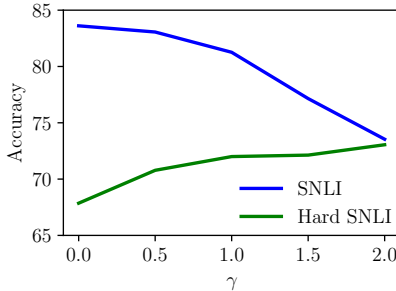
Figure 2: Accuracy of InferSent model trained with Debiased Focal Loss on SNLI and SNLI hard sets for different values of γ. Increasing γ reduces the contribution of the biased examples to the loss and concentrates the model on learning hard examples.

5 RELATED WORK

Biases in NLU benchmarks and other domains Recent studies have shown that large-scale NLU benchmarks contain biases. Poliak et al. (2018); Gururangan et al. (2018); McCoy et al. (2019) demonstrate that textual entailment models can rely on annotation artifacts and heuristic patterns to perform unexpectedly well. On ROC Stories corpus (Mostafazadeh et al., 2016), Schwartz et al. (2017) show that considering only sample endings without story contexts performs exceedingly well. A similar phenomenon is observed in fact verification (Schuster et al., 2019), argument reasoning comprehension (Niven & Kao, 2019), and reading comprehension (Kaushik & Lipton, 2018). Finally, several studies confirm biases in VQA datasets, leading to accurate question-only models ignoring visual content (Goyal et al., 2017; Zhang et al., 2016).

Existing techniques to alleviate biases The most common strategy to date to address biases is to augment the datasets by balancing the existing cues (Schuster et al., 2019; Niven & Kao, 2019). In another line of work, to address the shortcoming in Stanford Question Answering dataset (Rajpurkar et al., 2016), Jia & Liang (2017) propose to create an adversarial dataset in which they insert adversarial sentences to the input paragraphs. However, collecting new datasets especially in large-scale is costly and it remains an unsatisfactory solution. It is, therefore, crucial to develop strategies to allow training models on the existing biased datasets, while improving their out-of-domain performance.

Schuster et al. (2019) propose to first compute the n-grams existing in the claims which are the most associated with each label. They then solve an optimization problem to assign a balancing weight to each training sample to alleviate the biases. In contrast, we propose several end-to-end debiasing strategies. Additionally, Belinkov et al. (2019a) propose adversarial techniques to remove from the sentence encoder the features which allow a hypothesis-only model to succeed. However, we believe that in general the features used by the hypothesis-only model can include some information necessary to perform the NLI task, and removing such information from the sentence representation can hurt the performance of the full model. Their approach consequently degrades the performance on hard SNLI dataset which is expected to be less biased. In contrast to their method, we propose to train a bias-only model to use its predictions to dynamically adapt the classification loss to reduce the importance of the most biased examples during training.

Concurrently to our own work, Clark et al. (2019); He et al. (2019) have also proposed to use the product of experts models. However, we have evaluated on new domains and datasets, and have proposed several different ensemble-based debiasing techniques.

6 CONCLUSION

We propose several novel techniques to reduce biases learned by neural models. We introduce a bias-only model that is designed to capture biases and leverages the existing shortcuts in the datasets to succeed. Our debiasing strategies then work by adjusting the cross-entropy loss based on the performance of this bias-only model to focus learning on the hard examples and down-weight the importance of the biased examples. Our proposed debiasing techniques are model agnostic, simple and highly effective. Extensive experiments show that our methods substantially improve the model robustness to domain-shift, including 9.76 points gain on FEVER symmetric test set, 5.45 on HANS dataset and 4.78 points on SNLI hard set.

REFERENCES

Yonatan Belinkov, Adam Poliak, Stuart Shieber, Benjamin Van Durme, and Alexander Rush. Don't take the premise for granted: Mitigating artifacts in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019a.

Yonatan Belinkov, Adam Poliak, Stuart M Shieber, Benjamin Van Durme, and Alexander M Rush. On adversarial removal of hypothesis-only bias in natural language inference. In *Proceedings of the 8th Joint Conference on Lexical and Computational Semantics*, 2019b.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015.

Remi Cadene, Corentin Dancette, Hedi Ben-younes, Matthieu Cord, and Devi Parikh. Rubi: Reducing unimodal biases in visual question answering. In *Advances in neural information processing systems*, 2019.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. Enhanced lstm for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 2017.

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 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*, 2019.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017.

Ido Dagan, Oren Glickman, and Bernardo Magnini. The pascal recognising textual entailment challenge. *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, 2006.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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*, 2019.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A Smith. 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*, 2018.

He He, Sheng Zha, and Haohan Wang. Unlearn dataset bias in natural language inference by fitting the residual. 2019. URL https://arxiv.org/abs/1908.10763.

Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 2002.

Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, 2017.

Divyansh Kaushik and Zachary C Lipton. How much reading does reading comprehension require? a critical investigation of popular benchmarks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, 2017.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 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*, 2019.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016.

Lili Mou, Rui Men, Ge Li, Yan Xu, Lu Zhang, Rui Yan, and Zhi Jin. Natural language inference by tree-based convolution and heuristic matching. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 2016.

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

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, 2018.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018. URL `https://www.cs.ubc.ca/~amuham01/LING530/papers/radford2018improving.pdf`.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016.

Tal Schuster, Darsh J Shah, Yun Jie Serene Yeo, Daniel Filizzola, Enrico Santus, and Regina Barzilay. Towards debiasing fact verification models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019.

Roy Schwartz, Maarten Sap, Ioannis Konstas, Leila Zilles, Yejin Choi, and Noah A Smith. Story cloze task: Uw nlp system. In *Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics*, 2017.

Rishi Sharma, James Allen, Omid Bakhshandeh, and Nasrin Mostafazadeh. Tackling the story ending biases in the story cloze test. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, 2018.

Alex Wang, Amapreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*, 2019.

Adina Williams, Nikita Nangia, and Samuel Bowman. 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*, 2018.

Peng Zhang, Yash Goyal, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Yin and yang: Balancing and answering binary visual questions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

A FACT VERIFICATION

Base model: We finetune BERT for 3 epochs and use the default parameters and default learning rate of $2e-5$.

Bias-only model: Our bias-only classifier is a shallow nonlinear classifier with 768, 384, 192 hidden units with Tanh nonlinearity.

B TEXTUAL ENTAILMENT

Base model: InferSent uses a separate BiLSTM encoder to learn sentence representations for premise and hypothesis, it then combines these embeddings following Mou et al. (2016) and feeds them to the default nonlinear classifier. For InferSent we train all models for 20 epochs as default without using early-stopping. We use the default hyper-parameters and following Wang et al. (2019), we set BiLSTM dimension to 512. We use the default nonlinear classifier with 512 and 512 hidden neurons with Tanh nonlinearity. For Bert model, we finetune the models for 3 epochs.

Bias-only model For BERT model, we use the same shallow nonlinear classifier explained in Appendix A, and for the InferSent model, we use a shallow linear classifier with 512, and 512 hidden units.

C SYNTACTIC BIAS

Base model: We finetune all the models for 3 epochs.

Bias-only model: We use a nonlinear classifier with 6 and 6 hidden units with Tanh nonlinearity.

Results: Table 6 shows the performance for each label (entailment and non_entailment) on HANS dataset and its individual heuristics.

Table 6: Accuracy for each label of entailment or non-entailment on HANS and its individual heuristics.

Debiasing Method	gold label: Entailment				gold label: Non-entailment			
	HANS	Const.	Lexical	Subseq.	HANS	Const.	Lexical	Subseq.
None	98.37	98.98	96.76	99.38	23.82	23.24	41.18	7.04
RUBI+log space	97.51	98.56	95.44	98.54	27.55	16.98	54.92	10.76
RUBI	97.27	99.18	95.26	97.38	23.42	13.84	46.92	9.50
RUBI+normalize	97.87	98.48	96.32	98.80	21.55	13.00	43.70	7.94
Debiased Focal loss	96.41	97.66	92.92	98.66	33.57	27.18	55.98	17.56
Product of experts	96.08	98.38	93.52	96.34	37.02	30.20	61.70	19.16