Guide the Learner: Controlling Product of Experts Debiasing Method Based on Token Attribution Similarities

Ali Modarressi[1⋆] **Hossein Amirkhani**[2] **Mohammad Taher Pilehvar**[3]
[1] Center for Information and Language Processing, LMU Munich
[2] Computer and Information Technology Department, University of Qom, Iran
[3] Tehran Institute for Advanced Studies, Khatam University, Iran
amodaresi@cis.lmu.de
amirkhani@qom.ac.ir
mp792@cam.ac.uk

Abstract

Several proposals have been put forward in recent years for improving out-of-distribution (OOD) performance through mitigating dataset biases. A popular workaround is to train a robust model by re-weighting training examples based on a secondary biased model. Here, the underlying assumption is that the biased model resorts to shortcut features. Hence, those training examples that are correctly predicted by the biased model are flagged as being biased and are down-weighted during the training of the main model. However, assessing the importance of an instance merely based on the predictions of the biased model may be too naive. It is possible that the prediction of the main model can be derived from another decision-making process that is distinct from the behavior of the biased model. To circumvent this, we introduce a fine-tuning strategy that incorporates the similarity between the main and biased model attribution scores in a Product of Experts (PoE) loss function to further improve OOD performance. With experiments conducted on natural language inference and fact verification benchmarks, we show that our method improves OOD results while maintaining in-distribution (ID) performance.[1]

1 Introduction

Overfitting to the training data is a big obstacle in learning patterns that generalize to unseen data. Traditionally, this is diagnosed by monitoring the performance of a trained model on an in-distribution (ID) test set. However, a bigger challenge is when both the training and test data have the same non-generalizable patterns, emerged as spurious correlations between input features and output labels (Gardner et al., 2021). For instance, in the natural language inference (NLI) task, it is shown that the occurrence of some task-neutral words, like a negation in hypothesis, is highly correlated with a specific class (Gururangan et al., 2018). While high-capacity models can learn a generalized distribution of labels from the inputs, they are prone to spurious patterns, also known as dataset biases (Clark et al., 2019; He et al., 2019). A model could exploit these biases during fine-tuning, leading to a model that achieve high ID performance, while it is highly fragile in out-of-distribution (OOD) settings (Schuster et al., 2019; McCoy et al., 2020).

Besides trying to prevent these non-generalizable artifacts from entering the dataset (Liu et al., 2022), it is reasonable to seek for more robust learning methods. This has been the basis for a multitude of research works that encourage models to rely on truly generalizable patterns. Most of these methods are based on the assumption that the learning method will inevitably exploit biases if they are present in a training example (Clark et al., 2019; Sanh et al., 2020; Mahabadi et al., 2020; Utama et al., 2020; Ghaddar et al., 2021). Therefore, they discourage the main model from paying much attention to the examples which are correctly classified by a biased model. Recently, it is shown that this assumption is questionable in the way that for a significant number of cases, the main model does not follow the biased model in treating biased examples (Amirkhani and Pilehvar, 2021). Therefore, depriving the training algorithm from the examples which are detected to be biased is a waste of training data.

In this paper, we propose an alternative way to discard biased examples. Instead of considering the mere prediction of the biased model, we monitor the way the model processes each example by computing its attribution scores over the input tokens. With the resulting scores, we adjust the proportion of the loss function that is a cross-entropy loss (CE) versus a Product of Experts loss (PoE). If

⋆ Work done as a Master's student at Iran University of Science and Technology (IUST).

[1] Our code is freely available at: https://github.com/amodaresi/Debias_w_Saliencies

Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, pages 1954–1959
May 2-6, 2023

the attribution scores are similar between the main and biased models, the loss becomes a PoE loss where a correct prediction from the biased model down-weights the contribution of the corresponding example. In contrast, dissimilarity between the scores suggests a different behaviour from the biased model and leads to a CE loss that only considers the main model's prediction. Experiments on natural language inference and fact verification demonstrate that our method significantly outperforms previous approaches in terms of OOD performance while preserving its ID performance.

2 Methodology

In this section we explain our debiasing solution for a given classification task with dataset $\mathcal{D}\{\boldsymbol{x_i}, y_i\}_{i=1}^{N}$. For the i^{th} training example, we denote the input sequence of tokens as $\boldsymbol{x_i}$ and the gold label as $y_i \in \{1, 2, ..., Y\}$ where Y is the number of classes. Specifically, the goal is to train a model—we denote it as the *main model*—on the training dataset ($\mathcal{D}$) so that it can also perform well on OOD datasets that do not necessarily share the same biases. Following other well-known paradigms for identifying biases in the dataset, we first need to design or train a *biased model* that employs shortcut methods to complete the task (Mahabadi et al., 2020; Sanh et al., 2020; Utama et al., 2020).

In what follows, we will first review the Product of Experts (PoE) method for bias reduction, which is based solely on the outputs of the main and biased models. Then we present our contribution to developing a novel debiasing method that takes into account not only the models' outputs but also their attribution scores.

Product of Experts. In the original Product of Experts (PoE) solution, the loss is based on combining the predictions of the main and biased model: $\sigma(f_B(\boldsymbol{x_i^b}))$ & $\sigma(f_M(\boldsymbol{x_i}))$ (Hinton, 2002; Mahabadi et al., 2020). The summation of the log-softmaxes ($\sigma(.)$) of both models combine the distributions so that the main would focus less on biased examples:

$$f_C(\boldsymbol{x_i}; \boldsymbol{x_i^b}) = \log(\sigma(f_B(\boldsymbol{x_i^b}))) + \log(\sigma(f_M(\boldsymbol{x_i})))$$

The PoE loss is the cross-entropy loss over the summation shown above:

$$\mathcal{L}_{\text{PoE}}(\theta_M; \theta_B) = -\log(\sigma(f_C^{y_i}(\boldsymbol{x_i}; \boldsymbol{x_i^b}))) \quad (1)$$

While PoE does produce promising results, training the main model only based on the biased model's output can undermine some instances that could have been helpful and non-biased. As stated in Amirkhani and Pilehvar (2021), correct prediction of an instance by the biased model does not necessarily imply that the instance is biased, as the behaviour of the two models might differ.

2.1 PoE with Saliencies

To determine how similarly the main model and bias model behave, we must compute the attribution scores of the input tokens for both models. Therefore, after fine-tuning the biased model, we compute the saliencies $\boldsymbol{S}$ using a gradient-based approach according to the *gradient*×*input* method (Kindermans et al., 2016):

$$\boldsymbol{S}_i = \left\| \frac{\partial y_i}{\partial \boldsymbol{h}_i^0} \odot \boldsymbol{h}_i^0 \right\|_2 \quad (2)$$

This method is based on the gradient of the logit of the output prediction y_i with respect to the input embeddings $\boldsymbol{h}_i^0$. We also obtain the main model saliencies during training. By computing the saliencies for both bias and main models, it is possible to estimate the contribution of each input token in a training instance to both models' predictions. Therefore, we can compute the inter-model similarity—between the saliencies of the two models; for instance, using the cosine similarity metric:

$$\rho = \frac{\boldsymbol{S}_{\text{Main}} \cdot \boldsymbol{S}_{\text{Biased}}}{\|\boldsymbol{S}_{\text{Main}}\| \, \|\boldsymbol{S}_{\text{Biased}}\|} \quad (3)$$

Since the saliencies defined in 2 can only have positive values, ρ will always be a number between 0 and 1. This would provide a complementary metric that shows how similar the two models behave on a specific example. Therefore, we can modify the original PoE loss function that only incorporates the output predictions and include the inter-model similarity in the debiasing loss function:

$$\begin{aligned} \mathcal{L}_{\text{PoE+Sals}}(\theta_M; \theta_B) = \rho^* \mathcal{L}_{\text{PoE}}(\theta_M; \theta_B) \\ +\alpha(1-\rho^*)\mathcal{L}_{\text{CE}}(\theta_M) \end{aligned} \quad (4)$$

where $\mathcal{L}_{\text{CE}}(\theta_M)$ is the cross-entropy (CE) loss on the main model. We also define ρ^* that adjusts the inter-model similarity (ρ) based on the PoE loss using the following formulation:

$$\begin{aligned} \rho^* &= \rho^{\exp(-\beta \mathcal{L}_{\text{PoE}}(\theta_M; \theta_B))} \\ &= \rho^{\exp(\beta \log(\sigma(f_C^{y_i}(\boldsymbol{x_i}; \boldsymbol{x_i^b}))))} \\ &= \rho^{\sigma(f_C^{y_i}(\boldsymbol{x_i}; \boldsymbol{x_i^b}))^\beta} \end{aligned} \quad (5)$$

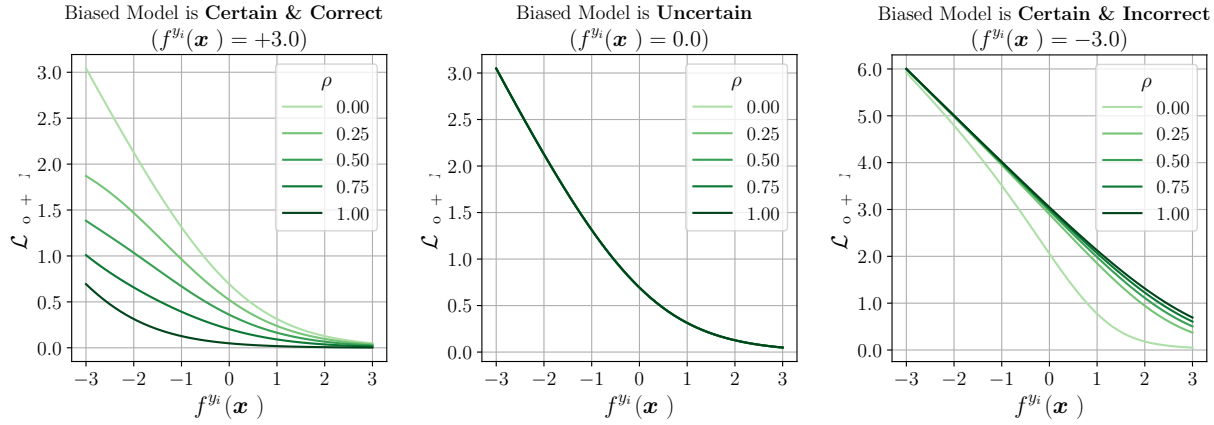

Figure 1: A visualization of our proposed loss function ($\mathcal{L}_{\text{PoE+Sals}}$) based on the main and biased models' prediction ($f_M^{y_i}(\boldsymbol{x_i})$ and $f_B^{y_i}(\boldsymbol{x_i})$) and the inter-model similarity (ρ). For a given input, when the biased model is correct and certain (left), the loss will be upweighted when the biased model differs from the main model in terms of saliency scores. However if the biased model returns a certain but incorrect prediction (right), the loss is downweighted if the main model is dissimilar and also returns a correct prediction. In an uncertain case (middle), the biased model does not affect the PoE loss (Sanh et al., 2020). In this case, the loss is converted to a CE loss based on the output of the main model.[3]

Where β is a positive hyperparameter for adjusting the combined prediction's ($\sigma(f_C^{y_i}(\boldsymbol{x_i}; \boldsymbol{x_i^b}))$) impact.

The intuition behind this adjustment is to upweight and increase the loss for an example where the main and biased models agree on the correct label but show a different behaviour in terms of token attribution scores (a.k.a. less inter-model saliency similarity scores, $0 < \rho < 0.5$). As an extreme case, if the biased model correctly classifies an example with a high output score ($\sigma(f_B^{y_i}(\boldsymbol{x_i^b})) \approx 1$), the main model's prediction would be ineffective in the original PoE loss, since the PoE model output probability is as follows:

$$\sigma(f_C^{y_i}(\boldsymbol{x_i}; \boldsymbol{x_i^b})) = \frac{\sigma(f_B^{y_i}(\boldsymbol{x_i^b}))\sigma(f_M^{y_i}(\boldsymbol{x_i}))}{\sum_{k=1}^{Y} \sigma(f_B^{k}(\boldsymbol{x_i^b}))\sigma(f_M^{k}(\boldsymbol{x_i}))}$$

Therefore, in the adjusted similarity stated in Eq. 5, the exponent would be approximately equal to one, which makes the adjusted similarity equal to the original cosine similarity. As a result, according to Eq. 4, if the models exhibit dissimilar explanations for their prediction ($\rho \approx 0$), the loss tends to be a cross-entropy loss[2] than a PoE loss. On the other hand, in case both models behave similarly on a given training example, i.e., $\rho \approx 1$, the example can be considered as one containing bias. Thus, the PoE loss renders this example as being less impactful during training. The left and right plots of Figure 1 respectively demonstrate[3] that having a higher similarity results in a loss function that is down- or up-weighted depending on the correctness of a certain biased model. Also with an uncertain biased model, the loss in Eq. 1 converts to a CE loss which is only based on the output of the main model.

3 Experiments

In this section, we will introduce the datasets, explain the experimental setup, and then demonstrate the results of our method.

3.1 Datasets

The experiments were carried out on two types of common NLU classification tasks: Natural Language Inference (NLI) and Fact Verification. For NLI, we used MNLI (Williams et al., 2018) as our in-distribution data and HANS (McCoy et al., 2020) for OOD evaluation. Note that the Matched development set is used for the ID evaluation in MNLI. In addition, because HANS has only two labels, entailment and not entailment, we consider outputs that are predicted contradictions or neutral as not entailment. For Fact Verification, we

[2]Weighted with α as a modulating hyperparameter ($0 < \alpha$).

[3]We simplify the plots by assuming the output logits of all classes except the gold label are zero for both models ($f_{B/M}^{y \neq y_i}(\boldsymbol{x_i}) = 0$). In addition, these figures are plotted based on $\alpha = 1$ and $\beta = 1$.

Model	MNLI		FEVER	
	Dev. (Matched)	HANS	Dev.	Sym.V1
BERT-base	**84.71±0.21**	62.85±2.69	85.19±0.37	56.51±1.41
DFL_{e2e} (Mahabadi et al., 2020)	83.91±0.20	66.10±2.81	80.48±0.87	65.13±1.52
PoE_{e2e} (Mahabadi et al., 2020)	84.10±0.19	63.63±1.90	84.43±0.87	64.28±1.52
PoE (Sanh et al., 2020)	81.10±0.41	68.04±1.51	80.08±0.94	62.72±2.99
PoE+CE (Sanh et al., 2020)	83.34±0.33	66.56±0.66	85.29±1.25	62.55±2.46
PoE w. Attribution Similarity	82.81±0.26	**68.06±0.66**	85.48±1.09	**66.97±2.00**
w/o Attribution Similarity	-	-	**86.09±0.72**	65.18±2.25

Table 1: The mean and standard deviation of the accuracy scores of multiple debiasing strategies applied to NLI and fact verification. The maximum values are highlighted in bold. It is important to note that the results of the methods that are used for comparison are not the results that were reported by the methods themselves but rather the scores that were obtained by replicating their implementation[5].

used FEVER (Thorne et al., 2018) and FEVER Symmetric-V1 (Schuster et al., 2019) for ID and OOD, respectively.[4]

3.2 Setup

In all the experiments, we used *BERT-base-uncased* (Devlin et al., 2019) as the main model to allow a fair comparison against other debiasing methods. However, the biased model differs depending on the task.

We used the TinyBERT model (Turc et al., 2019) for the NLI task, since its limited capacity makes it extremely susceptible to biased features in training examples (Sanh et al., 2020). For fact verification, a full 12-layer BERT-base model similar to the main model is fine-tuned using only the claim sentences from the training data. In both biased models, we compute and save their prediction logits and attribution scores across the entire training dataset so that they can be used as a frozen model during training. However, since the fact verification biased model is only trained on the claim sentences, the attribution scores are only calculated for the claim portion. Therefore, the similarities computed during the training procedure are limited to the claim segment alone.

For the generic fine-tuning hyperparameters, we adopted Sanh et al. (2020) recommendations: 3 epochs of training, a batch size of 32, an Adam optimizer (Kingma and Ba, 2015) with warmup and linear decay in its learning rate schedule, and a peak learning rate of 3e-5 or 2e-5 for MNLI or FEVER, respectively. But as for the specific hyperparameters in our approach, using sweeping over $\alpha \in \{0.01, 0.1, 0.2, 0.3, 0.5, 1.0\}$ and $\beta \in \{0.1, 0.3, 0.5, 1.0\}$, we set $\alpha = 1.0$, $\beta = 1.0$ for MNLI and $\alpha = 0.3$, $\beta = 0.1$ for training on FEVER.

All experiments were implemented using the HuggingFace Transformers library (Wolf et al., 2020) and performed on an RTX 3070 GPU machine. The results are the average of six runs with different seeds.

3.3 Results

The results of various debiasing techniques applied to the previously mentioned benchmarks are shown in Table Table 1. The baseline is fine-tuning the backbone model (BERT-base-uncased) with the commonly used cross-entropy loss, which provides high ID performance but lacks OOD. We also include four approaches from two different studies for additional comparison[5]. Two End-to-End solutions from Mahabadi et al. (2020), one utilizing Debiased Focal Loss (DFL) and the other employing PoE. The PoE and PoE+CE (PoE with a static weighted CE loss added) methods from Sanh et al. (2020) are similar to our approach of having a bi-

[4]In both FEVER and FEVER-Symmetric, replacement tokens are used in place of parentheses and brackets (e.g. "]" → "-RSB-"). This causes the BERT tokenizer to split the specified tokens into multiple segments, as there are no tokens in the BERT vocabulary that correspond to the replacement tokens. Therefore, we replace these tokens with the punctuation that corresponds to them so that BERT can tokenize the inputs with less undesirable segmentation. We apply this modification to other approaches that we have evaluated and compared.

[5]We could have included the compared methods' results from their respective papers as well as results from other approaches. However, in our empirical results we observed significant discrepancies between the results obtained from their source code and those reported. In this paper, we intend to report only the reproduced results, which is why the number of compared methods is limited.

ased model that is frozen, but they only rely on the predictions of the biased model and not the inter-model similarities.

It can be observed that the OOD performance of our method outperforms those of other approaches, without or with minimal loss of ID accuracy. On FEVER, our strategy improves SymV1 by nearly 2% while also slightly improving the ID performance. In MNLI, our strategy improves HANS to achieve high scores comparable to Sanh et al. (2020) PoE-only solution. However, the ID performance of Sanh et al. (2020) PoE-only falls short. While maintaining the same level of OOD performance, our solution achieves a minimum MNLI-m dev score that is acceptable.

As an ablation study, we also trained the procedure without using similarity values, resulting in a PoE+CE solution. Since we used a different biased model in FEVER than Sanh et al. (2020), we reported its results using the claim-only biased model configuration. However, we omitted MNLI results from our report because they are identical to Sanh et al. (2020) PoE+CE configuration. Having a claim-only biased model for FEVER rather than a TinyBERT model yields a substantial increase in OOD, as can be seen in the results. As a result, we can observe that even though adding a weighted CE to a PoE loss yields improvements in terms of ID performance in particular, the similarities could push even further and also improve OOD performance.

Another observation is the large standard deviations in OOD accuracy of all approaches, which is why we chose to execute 6 runs as also suggested by Sanh et al. (2020). Even so, it should be noted that our method still has a relatively low variance in HANS.

4 Conclusions

In this paper, we introduced a strategy for improving OOD performance by incorporating the similarity values between the token attribution scores of main and biased models into a Product of Experts (PoE) loss function. The gist of this approach is that it takes into account the decision-making process in addition to the output predictions of the main and biased models (which are often taken as the primary signal). By comparing our method to multiple recent debiasing methods on two widely-used NLU tasks, we demonstrated that our method improves performance on out-of-distribution data while preserving performance on in-distribution data. Future work could include exploring other loss formulations with Attribution Similarities that may produce better results. Additionally, it may be beneficial to investigate methods to reduce the variability of results, which is present in the majority of debiasing solutions.

Limitations

Due to the high variance in the accuracy scores, the main limitation of this approach (as well as the majority of other debiasing approaches) is the large number of seeds used to tune the hyperparameters. As a result, any type of tuning necessitates a multi-seed run, which requires multiple GPUs or many hours of training. Because of this constraint, we omitted working on larger scaled models or other types of PLMs.

References

Hossein Amirkhani and Mohammad Taher Pilehvar. 2021. Don't discard all the biased instances: Investigating a core assumption in dataset bias mitigation techniques. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4720–4728.

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.

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.

Matt Gardner, William Merrill, Jesse Dodge, Matthew E Peters, Alexis Ross, Sameer Singh, and Noah A Smith. 2021. Competency problems: On finding and removing artifacts in language data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1801–1813.

Abbas Ghaddar, Philippe Langlais, Mehdi Rezagholizadeh, and Ahmad Rashid. 2021. End-to-end self-debiasing framework for robust nlu training. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1923–1929.

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.

Geoffrey E. Hinton. 2002. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800.

Pieter-Jan Kindermans, Kristof Schütt, Klaus-Robert Müller, and Sven Dähne. 2016. Investigating the influence of noise and distractors on the interpretation of neural networks. *arXiv*, abs/1611.07270.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Alisa Liu, Swabha Swayamdipta, Noah A Smith, and Yejin Choi. 2022. Wanli: Worker and ai collaboration for natural language inference dataset creation. *arXiv preprint arXiv:2201.05955*.

Rabeeh Karimi Mahabadi, James Henderson, et al. 2020. End-to-end bias mitigation by modelling biases in corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, CONF.

R Thomas McCoy, Ellie Pavlick, and Tal Linzen. 2020. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *57th Annual Meeting of the Association for Computational Linguistics, ACL 2019*, pages 3428–3448. Association for Computational Linguistics (ACL).

Victor Sanh, Thomas Wolf, Yonatan Belinkov, and Alexander M Rush. 2020. 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.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. 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 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: On the importance of pre-training compact models. *arXiv preprint arXiv:1908.08962v2*.

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

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.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.