Exploration of Methods to Remove Bias From Models Performing Natural Language Inference Task

Ilyana Anderson

Abstract

In this project, we[1] make several attempts to remove bias from the models trained for the natural language inference task. We explore the possibility that a model forced to generate a hypothesis based on a premise might be less prone to learning the dataset bias. Another attempt to reduce the learned bias was made by flipping premises and hypotheses for the contradiction examples during the training. Lastly, we implemented the DRiFT method reported in the literature. Most of the enhanced models demonstrated improved performance on the HANS dataset relative to the baseline models, suggesting a potential reduction in bias. Nevertheless, while these results are indicative, they are not conclusive enough to firmly assert the complete removal of bias.

1 Introduction

Natural language inference (NLI) is a task designed to measure the understanding of the relationship between two sentences. The relationship could be that of *entailment* if the information from one sentence (hypothesis) appears to be true based on the information from the other sentence (premise). Other possible relationships are those of *neutrality* and *contradiction*. Each premise-hypothesis pair carries a label: entailment, neutral, or contradiction. The hope behind the construction of this task is to force the model to develop a deep understanding of the implications of sentences.

Two benchmark datasets were developed to measure the performance on NLI task: Stanford Natural Language Inference (SNLI) dataset (Bowman et al., 2015) and the Multi-Genre Natural Language Inference (MNLI) dataset (Williams et al., 2017). The current models pre-trained on large text corpora can achieve high accuracy on these datasets. However, with this achievement, a question arises whether the models gained sufficient understanding of the language or they tend to rely on other signals. *Dataset bias* (Torralba and Efros, 2011; He et al., 2019) is understood as a set of superficial clues present in a dataset. It is easy for a model to learn these clues instead of the more relevant information.

The effects of the learned bias could be observed in multiple ways. First, models do not generalize well across datasets or in an out-of-distribution setting. It will be shown in the current work that models trained on the SNLI dataset have a drop in accuracy when tested on the MNLI dataset and vice versa. The bias could also manifest itself in the model's ability to make a decision based on the hypothesis alone (Poliak et al., 2018). One more example of the bias seen in a trained model is its tendency to rely on certain types of syntactic heuristics when predicting an *entailment* label (McCoy et al., 2019).

This project aims to explore three distinct approaches for mitigating learned bias in models through varied training methodologies applied to the same dataset. Our findings show that the enhanced models generally exhibit improved performance on the HANS dataset (McCoy et al., 2019). Nonetheless, it is observed that these debiased models face challenges in achieving similar efficacy on other adversarial datasets, thereby precluding a definitive determination of the effectiveness of the proposed methods.

2 Baseline Models Performance and Problem Statement

2.1 In-distribution and out-of-distribution performance

Three pre-trained models were further fine-tuned on the NLI task: BART (Lewis et al., 2019),

[1]This is a one-person project. The plural pronoun was chosen to adhere to the standards of academic writing.

BART (after 3 epochs)

trained on SNLI			trained on MNLI		
in-distribution	out-of-distribution (mnli)	% drop	in-distribution	out-of-distribution (snli)	% drop
93.2%	84.6%	-8.6%	90.1%	88.5%	-1.6%

BERT (after 3 epochs)

trained on SNLI			trained on MNLI		
in-distribution	out-of-distribution (mnli)	% drop	in-distribution	out-of-distribution (snli)	% drop
90.0%	71.3%	-18.7%	84.1%	78.9%	-5.2%

ELECTRA (after 4 epochs)

trained on SNLI			trained on MNLI		
in-distribution	out-of-distribution (mnli)	% drop	in-distribution	out-of-distribution (snli)	% drop
88.3%	69.4%	-18.9%	79.8%	73.2%	-6.6%

Table 1: Performance of baseline models on in-distribution and out-of-distribution validation data.

BERT (Devlin et al., 2018), and ELECTRA (Clark et al., 2020). The following variants of these models were taken from Huggingface Transformers library: facebook/bart-large, bert-base-uncased, and google/electra-small-discriminator. Each pre-trained model was further trained on SNLI and on MNLI datasets, producing two separate fine-tuned models. We used the following parameters during the training:

- Maximum allowed number of tokens in an example: this value was set to the default value for each corresponding tokenizer (512 for ELECTRA and BERT, 1024 for BART)
- Batch size: 256
- Learning rate main value: 2e-5
- Learning rate schedule: 200 warmup steps that bring the learning rate to the main value (corresponds to 51,200 examples), then linear decay afterwards.
- Optimizer: Adam optimizer with weight decay fix (Loshchilov and Hutter, 2017), used as AdamW optimizer in Huggingface with default parameters.

Each of the models was trained for 4 epochs. At the end of each epoch, we evaluated the performance on the in-distribution validation dataset (validation set for SNLI and "matched" validation set for MNLI) and on the corresponding out-of-distribution validation dataset. For out-of-distribution evaluation, we used MNLI "matched" validation set for models trained on SNLI and SNLI validation set for models trained on MNLI.

For some of the models, we observed a drastic increase of loss values on the in-distribution and out-of-distribution validation sets. This increase might be indicative of overfitting to the training data and the degradation of the model. When choosing the best-performing models, we considered both the accuracy and the loss on the in-distribution and out-of-distribution validation sets. In other words, we aimed for the models to have high accuracy values while maintaining reasonably low loss values on the validation sets. After this consideration, we chose BART and BERT models after 3 epochs of training and ELECTRA after 4 epochs of training as our best-performing models. The results for the best-performing models are shown in Table 1. The column labeled "% drop" indicates the drop of percentage points from in-distribution to out-of-distribution validation accuracy.

We will use these models as the baseline models that we would like to analyse in this section. For convenience, we label the baseline models the following way:

Label	Description
bart_snli_3e	BART, trained on SNLI for 3 epochs
bart_mnli_3e	BART, trained on MNLI for 3 epochs
bert_snli_3e	BERT, trained on SNLI for 3 epochs
bert_mnli_3e	BERT, trained on MNLI for 3 epochs
electra_snli_4e	ELECTRA, trained on SNLI for 4 epochs
electra_mnli_4e	ELECTRA, trained on MNLI for 4 epochs

We observe that a larger model (BART) in

Model	accuracy on entailment	accuracy on non-entailment	overall accuracy
bart_snli_3e	98.7%	33.9%	66.3%
bart_mnli_3e	99.7%	48.0%	73.9%
bert_snli_3e	99.2%	5.6%	52.4%
bert_mnli_3e	99.0%	4.2%	51.6%
electra_snli_4e	99.6%	0.4%	50.0%
electra_mnli_4e	95.7%	10.6%	53.1%

Table 2: Performance of baseline models on HANS validation dataset.

general achieves higher accuracy on both in-distribution and out-of-distribution datasets, with the minimal drop of percentage points. However, all models investigated in the present work exhibit a drop of percentage points between in-distribution and out-of-distribution performance. It indicates the models' struggle to generalize across datasets.

We also observe that the models trained on MNLI achieve lower in-distribution accuracy compared to the models trained on SNLI. This might indicate the higher difficulty level of MNLI validation data compared to SNLI validation data.

Additionally, models trained on MNLI exhibit lower percentage drops. Various factors could be contributing to this result. First, the relative difficulty level of SNLI and MNLI validation data could be responsible. Second, models trained on MNLI could be more robust compared to the models trained on SNLI. Further accuracy measurements on multiple sets of adversarial data are needed to confirm this possibility. However, this is outside of the scope for this project.

2.2 Performance on HANS dataset

Jia (2017), Wallace (2019), Gardner (2020), and others have shown that models tend to struggle with performance on adversarial data designed to target possible weaknesses of models' "reasoning" on various NLP tasks.

In this work, we test our baseline models on Heuristic Analysis for NLI Systems (HANS) dataset (McCoy et al., 2019).

HANS dataset was designed to target the general tendency of models to rely on several possible syntactic heuristics when predicting *entailment* versus *non-entailment* labels: the lexical overlap heuristic, the subsequence heuristic, and the constituent heuristic. Examples from each type of heuristic are:

- Lexical Overlap (non-entailment): *"The president advised the doctor."* and *"The doctor advised the president."*
- Lexical Overlap (entailment): *"The actors that the students contacted admired the lawyer."* and *"The students contacted the actors."*
- Subsequence (non-entailment): *"The managers investigated in the laboratory recognized the professor."* and *"The managers investigated in the laboratory."*
- Subsequence (entailment): *"The scientists saw the artists who the author recognized."* and *"The scientists saw the artists."*
- Constituent (non-entailment): *"Supposedly the professors called the judges."* and *"The professors called the judges."*
- Constituent (entailment): *"Obviously the artists ran."* and *"The artists ran."*

Since the dataset labels are *entailment* and *non-entailment*, we had to make a choice about how to calculate the accuracy for our baseline models, which predict three labels (*entailment*, *neutral*, and *contradiction*).

Given the information about how this dataset was constructed and visually inspecting the examples from the dataset, our original choice was to equate *non-entailment* label to *neutral*.

However, the creators of the dataset argue that in some examples (e.g. *"The actor was helped by the judge."* and *"The actor helped the judge."*) a non-strict *contradiction* label could be assigned. Furthermore, they examined two ways to calculate the accuracy for non-entailment examples in the dataset. One approach was to assign a label normally, then collapse *neutral* and *contradiction* labels into *non-entailment*. The second approach was to add logits corresponding to *neutral* and

Model	accuracy on custom dataset
bart_snli_3e	80.6%
bart_mnli_3e	75.0%
bert_snli_3e	61.1%
bert_mnli_3e	61.1%
electra_snli_4e	52.8%
electra_mnli_4e	47.2%

Table 3: Performance of baseline models on the custom dataset designed to target Hypothesis Alone bias.

contradiction labels prior to assigning a label. It was reported (McCoy et al., 2019) that these two methods yield similar results, so in this work we use the first method of collapsing *neutral* and *contradiction* labels into *non-entailment*.

The results for our baseline models on HANS validation dataset are presented in Table 2.

We observe a relatively low accuracy for non-entailment examples for all our models, with values ranging from 48.0% to as low as 0.4%. It indicates that all our baseline models learned to rely on the syntactic heuristics when making a prediction, and thus, they learned a certain type of bias during their training. In this work, we will refer to this bias as the Entailment bias.

2.3 Performance on the custom dataset designed to target Hypothesis Alone bias

For the NLI task, models could pick up signals present in a dataset that would enable them to make a correct prediction based on the information from one sentence instead of a pair of sentences. One example of such undesirable clue is a positive correlation between negation and a contradiction label: e.g., a hypothesis such as *The woman did not board the train* is more likely to carry a label of *contradiction* rather than *entailment* or *neutral* in a premise-hypothesis pair. This correlation likely arose owing to the dataset creation strategy (Poliak et al., 2018). In this report we will refer to this kind of bias as the Hypothesis Alone bias.

To test for the presence of the Hypothesis Alone bias, we have created a small dataset (36 examples) in which hypothesis sentences contain a negation word (such as *not*) in examples labeled as *entailment* or *neutral*. The examples labeled as *contradiction* are free from negation words in this dataset.

As an example, for the premise *"The man is playing music"*, we wrote three possible hypotheses: *"The woman does not sing well"* (label: *neutral*), *"The man is not scuba-diving"* (label: *entailment*), *"The man is dead"* (label: *contradiction*).

The accuracy values on the custom dataset for our baseline models are given in Table 3.

Based on the results reported in the previous subsections, we expected that the models trained on MNLI would outperform the models trained on SNLI. However, the result we observed is the opposite. We hypothesize that this might be due to the fact that the sentences from our custom dataset are closer to the sentences from SNLI in length. However, this dataset is too small for any strong conclusions to be drawn.

Overall, the baseline models achieve low to medium accuracy on the custom dataset, with values ranging from 47.2% to 80.6%. The majority baseline for this dataset is 38.9%.

2.4 Performance in Hypothesis Alone setting

The following accuracy results were reported by Poliak (2018) on SNLI validation and MNLI "matched" validation datasets for the models trained to classify just the hypotheses:

SNLI	67.17%
MNLI	55.52%

In the present work, we are interested whether the baseline models trained normally (i.e. seeing unmodified data, both a premise and a hypothesis, in each training example) can make correct predictions when presented with the input in which any useful information could be derived only from the hypothesis. We call this setting a Hypothesis Alone setting.

Three methods to modify data in order to present the input in a Hypothesis Alone setting were investigated, and the results are presented in Table 4.

Firstly, we replaced all premises by an empty string "" (Empty String method). As the second

Model	Empty String	Single Sentence	Attention Mask	Majority Baseline
bart_snli_3e	44.4%	48.6%	-	33.8%
bart_mnli_3e	40.1%	35.7%	-	35.5%
bert_snli_3e	50.7%	46.0%	50.4%	33.8%
bert_mnli_3e	36.8%	36.5%	38.0%	35.5%
electra_snli_4e	49.5%	49.1%	46.5%	33.8%
electra_mnli_4e	38.0%	38.2%	38.1%	35.5%

Table 4: Performance of baseline models in the Hypothesis Alone setting. Hypothesis Alone setting was implemented by various methods (Empty String, Single Sentence, and Attention Mask). The highest result for each model is shown by color. The majority baseline for SNLI and MNLI validation data is given for the reference.

method, we changed the data format from two-sentence format (when the premise and hypothesis are separated by a separator token) to one-sentence format (only one string is fed as an input into the model). The second method is presented in Table 4 as a Single Sentence method. As the third method (implemented for BERT and ELECTRA models), we masked out the premises so that all tokens in a premise sentence would be replaced by padding tokens (Attention Mask method). The evaluation was performed on the corresponding in-distribution dataset for each model (validation SNLI dataset for models trained on SNLI and "matched" validation MNLI dataset for models trained on MNLI).

The results shown by Poliak (2018) indicate that compared to MNLI, SNLI dataset exhibits stronger signal that could be exploited to make a correct prediction based on the hypothesis alone. Our measurements confirm this finding.

In future sections, we will test our enhanced models in the Hypothesis Alone setting. For each model, we pick only one method of implementing the Hypothesis Alone setting (Empty String, Single Sentence, or Attention Mask). More specifically, we pick the method for which the corresponding baseline model shows the highest accuracy: for example, if the baseline model is BART trained on SNLI, Single Sentence method would be picked. Our goal for the enhanced models would be to lower this number.

2.5 Problem Statement

We confirmed that models of various sizes trained on SNLI and on MNLI data using the standard maximum likelihood estimation (MLE) exhibit two biases: Entailment bias and Hypothesis Alone bias. The bias is most pronounced in smaller models (ELECTRA and BERT) although all six of our baseline models are biased. We have further shown that the models trained normally on premise-hypothesis pairs of SNLI dataset can still make relatively accurate predictions based on hypothesis alone.

2.6 Possible Goals

Our exploration of the ways to remove learned bias was conducted with the following possible goals in mind: increase of accuracy on out-of-distribution datasets, on HANS dataset, or on the custom dataset. Additionally, we were interested to determine whether the accuracy in Hypothesis Alone setting would go down upon an execution of an attempt to remove Hypothesis Alone bias.

3 Generating Hypothesis as a possible way to remove bias (models: Enhanced 1 and Enhanced 2)

The Hypothesis Alone bias could arise if the model "pays too much attention" to the hypothesis sentence, possibly ignoring the information coming from the premise. Our intuition was that by forcing the model to rely heavily on the information from the premise, the Hypothesis Alone bias might be removed.

Additionally, we believe that it might be possible to use the training techniques applicable to humans on the neural network models. For the majority of language-related tasks, humans are encouraged to generate their own text in the process of learning. In this experiment we want the model to generate its own text (the hypothesis sentence) as well.

Our idea was to first train a model to generate a hypothesis sentence based on the premise. The model would have to heavily rely on the information from the premise in order to perform this task. The weights of the model trained this way would

Premise: "Under a blue sky with white clouds, a child reaches up to touch the propeller of a plane standing parked on a field of grass."	
After 1 epoch	**After 2 epochs**
Hypothesis generated for the *neutral* label: "A child reaches up to touch the propeller of a plane parked on a field of grass." (This generated hypothesis is wrong for the *neutral* label.)	Hypothesis generated for the *neutral* label: "A child reaches up to touch the propeller of a plane parked on a field of grass on a sunny day." (This generated hypothesis could be judged as wrong or correct for the *neutral* label. It is better because new information about sunny day was added.)
Hypothesis generated for the *contradiction* label: "Under a blue sky with white clouds, a child reaches up to touch the propeller of a car parked on a field of grass."	Hypothesis generated for the *contradiction* label: "The child is flying a plane." (This sentence was deemed to be a better *contradiction* sentence because it involved new action of flying a plane as opposed to a new object, the car. It is our assumption that generating a new action is more difficult than generating a new object.)

Table 5: Performance of BART model trained on SNLI data on the sequence-to-sequence language modeling task, together with sample evaluation.

be used to further fine-tune the model on the NLI task.

3.1 Implementation Details

The baseline model for these experiments is BART model trained on SNLI. In Section 2, we presented results for this model after epoch 3. However, since the accuracy values vary from epoch to epoch on various datasets, we show the results for all epochs for this model in Table 6. The highest accuracy value achieved on a specific dataset across all epochs is taken to be the baseline value for these experiments (note: for the performance in Hypothesis Alone setting, the lowest value is taken to be the baseline value because we generally do not want debiased models to predict the label based on the hypothesis alone).

A large model (BART) was chosen for the relative ease of implementing the training on a sequence-to-sequence language modeling task. SNLI dataset was chosen because it exhibits a higher Hypothesis Alone bias compared to MNLI, as was shown in Section 2.

The dataset was modified in the following way: for each example, a premise was attached to a command. Three commands were used, depending on the label: *Generate an entailment sentence, Generate a contradicting sentence, Generate a neutral sentence*. The hypothesis was used as a target sentence that the model needed to generate.

The hyperparameters for the training are the same as for the baseline models.

The model was trained for 4 epochs and saved after each epoch. We evaluated the results after each epoch on SNLI validation data modified accordingly to fit the sequence-to-sequence language modeling task. Originally, we used BLEU and ROUGE scores for evaluation, however, these scores were comparable for trained and untrained models and thus did not provide useful information. As an alternative, we evaluated the results by comparing the generated outputs after each epoch visually and using our judgement about which output was of better quality. An example of the generated output as well as our evaluation is presented in Table 5. Based on our evaluation we chose the model after 2 epochs as the best-performing sequence-to-sequence (seq2seq) model. In general, even the best-performing model sometimes struggled to generate sentences for the *neutral* label.

After we obtained the best-performing seq2seq model, we substituted the seq2seq language modeling head by the sequence classification head from Huggingface library and further fine-tuned the model on NLI task for 4 epochs on SNLI dataset. We called the resulting model Enhanced 1. The results for Enhanced 1 model across all epochs are given in Table 7.

One additional experiment was conducted after obtaining the best-performing seq2seq model. This experiment is similar to how we trained Enhanced 1 model, but we froze all layers in the

Evaluation Data	1 epoch	2 epochs	3 epochs	4 epochs
snli	92.5%	93.0%	93.2%	93.3%
mnli	83.9%	84.7%	84.6%	84.3%
hans_entailment	94.8%	99.0%	98.7%	97.8%
hans_non-entailment	38.7%	36.0%	33.9%	36.5%
hans_non-ent (non-ent = neutral)	0.6%	1.0%	2.1%	3.6%
custom	72.2%	75.0%	80.6%	75.0%
hypot_alone (single sent.)	48.4%	47.6%	48.6%	46.8%

Table 6: Performance of BART model trained on SNLI data across 4 epochs on NLI task (baseline model). The highest accuracy values across epochs are highlighted in pink and the lowest in blue. Values highlighted in color are taken to be the baseline accuracy values for Enhanced 1 model (see Table 7).

Evaluation Data	1 epoch	2 epochs	3 epochs	4 epochs	Baseline (best result)
snli	92.7%	93.0%	93.1%	93.1%	93.3%
mnli	84.0%	84.4%	84.5%	84.4%	84.7%
hans_entailment	97.6%	97.5%	98.2%	97.8%	99.0%
hans_non-entailment	45.4%	45.5%	45.3%	51.1%	38.7%
hans_non-ent (non-ent = neutral)	2.7%	3.0%	5.3%	7.1%	3.6%
custom	77.8%	75.0%	80.6%	75.0%	80.6%
hypot_alone (single sent.)	48.2%	47.2%	47.9%	46.3%	46.8%

Table 7: Results for Removal of Bias by Generating Hypothesis experiment across 4 epochs (model: Enhanced 1). The highest accuracy values across epochs are highlighted in pink and the lowest in blue. The baseline values are given for the reference on the right: they are the best result values for the baseline model across 4 epochs.

Evaluation Data	Enhanced 2 after 1 epoch
snli	92.0%
mnli	82.1%
hans_entailment	90.5%
hans_non-entailment	47.6%
hans_non-ent (non-ent = neutral)	0.7%
custom	69.4%
hypot_alone (single sent.)	41.3%

Table 8: Results for Removal of Bias by Generating Hypothesis experiment after 1 epoch (model: Enhanced 2).

encoder part of the model. The resulting model is Enhanced 2. The experiment was supposed to run for 4 epochs, but unfortunately the training crashed after 1 epoch. Due to time limitations, we did not re-run the experiment, and thus we have only one model (the model after 1 epoch) for Enhanced 2. The results for for Enhanced 2 are presented in Table 8. The performance of Enhanced 2 model is worse that the performance of Enhanced 1 model (after 1 epoch) on all datasets except HANS non-entailment examples. The freezing of the layers could have had a negative impact on the performance. However, no definitive conclusions should be drawn from this model because it has not completed 4 epochs of training.

3.2 Results and Discussion (model Enhanced 1 as compared to the baseline model)

The results for this experiment were surprising to us. Since we targeted Hypothesis Alone bias, we expected improved results on the out-of-distribution (MNLI) dataset and on the custom dataset. However, the improvement of the accuracy can be observed only on HANS dataset (on non-entailment examples), which is designed to target Entailment bias.

Similar results were reported by He (2019). When they attempted to remove bias in three different ways, a drastic improvement was observed for all BERT models on HANS dataset. For other datasets that they used for evaluation, the improve-

ment was not as obvious as on HANS.

We hypothesize that the improvement on HANS dataset might be due to a better "reasoning" acquired by the model after the removal of any undesirable bias, regardless of the type of the bias.

Another possible reason might be the lack of clear-cut right or wrong label within the dataset itself for the case of *non-entailment* labels. HANS dataset was designed to carry two labels while the models are predicting three. So, if the model is predicting *contradiction* label for a *non-entailment* HANS label, we count it as correct. However, in the strictest sense, *non-entailment* does not mean *contradiction*. As a result, the increase of accuracy on HANS non-entailment examples constitute weak evidence that the model is improving.

We have carried out additional evaluation on HANS dataset (non-entailment examples), in which we equated *non-entailment* label to *neutral* (in this case, predicting *contradiction* is counted as wrong). In tables, this evaluation is presented as "hans_non-ent (non-ent = neutral)". In this evaluation, the accuracy values of Enhanced 1 improved compared to the accuracy values for the baseline model. However, since the absolute increase of percentage points is low and the values vary significantly from epoch to epoch, no strong conclusions should be drawn.

4 Switching premises and hypothesis for contradiction examples as a possible way to remove bias (models: Enhanced 3 and Enhanced 4)

Similar to the previous section, in this section our goal is to remove Hypothesis Alone bias. Our intuition for these experiments remains the same: we believe that the Hypothesis Alone bias could arise if the model "pays too much attention" to the hypothesis sentence, which is a second sentence in a premise-hypothesis pair.

By visual inspection of *contradiction* examples in MNLI and SNLI dataset, we arrived at the conclusion that premises and hypothesis could be switched for those examples without compromising the correctness of the label. We will call these examples "flipped". If a model sees "flipped" examples in the dataset instead of normal *contradiction* examples, it might avoid picking up the undesired signal from the hypotheses because the ordering of the sentences is flipped.

4.1 Implementation Details

Two experiments were performed, with the resulting models Enhanced 3 and Enhanced 4.

The baseline model for Enhanced 3 is ELECTRA model trained on SNLI data. In Section 2, we presented results for this model after 4 epochs of training. The accuracy values after each epoch can be seen for this baseline model in Table 9.

To obtain Enhanced 3 model, we trained ELECTRA model on SNLI data in the same way as the baseline model, but with data modification: the premises and hypotheses for *contradiction* examples were flipped. Two variants of baseline values were considered for this model. First, we have taken the "best" values for the baseline model across all 4 epochs. Second, we considered the baseline model after 4 epochs as the best-performing model in general, so we have taken the exact results after 4 epochs of training as the baseline values. Both baselines are presented in Table 10 together with the measurements for Enhanced 3.

The baseline values for Enhanced 4 is BART evaluated on various datasets after 2 epochs of training on SNLI (see Table 6).

To obtain Enhanced 4, we first trained BART normally on SNLI for 1 epoch. Then we took the ad interim model and further trained it for one more epoch on SNLI, with data modification: the premises and hypotheses for *contradiction* examples were flipped. Since the model was trained for 2 epochs in total on SNLI dataset (first unmodified, then modified), it was determined that the baseline values for this experiment should be the exact values of the baseline model after 2 epochs of training as well. The evaluation of Enhanced 4 is presented in Table 11.

4.2 Results and Discussion

We observe higher accuracy on HANS non-entailment examples for Enhanced 3 and Enhanced 4. For Enhanced 3, this increase is seen only for the less strict measurement of accuracy when both *neutral* and *contradiction* labels are collapsed into *non-entailment*. The reasons why the enhanced models might be performing better on HANS non-entailment examples were discussed in Section 3, and they remain possible for these experiments as well.

We also observe that Enhanced 3 lost its ability to predict correct label based on the hypothesis

Evaluation Data	1 epoch	2 epochs	3 epochs	4 epochs
snli	86.1%	87.7%	88.3%	88.3%
mnli	66.6%	68.7%	69.7%	69.4%
hans_entailment	99.9%	99.4%	99.6%	99.6%
hans_non-entailment	0.1%	0.2%	0.2%	0.4%
hans_non-ent (non-ent = neutral)	0.04%	0.01%	0.02%	0.02%
custom	44.4%	47.2%	52.8%	52.8%
hypot_alone (empty string)	49.5%	49.8%	50.2%	49.4%

Table 9: Performance of ELECTRA model trained on SNLI data across 4 epochs on NLI task (baseline model for Enhanced 3 experiment). The highest accuracy values across epochs are highlighted in pink and the lowest in blue.

Evaluation Data	1 epoch	2 epochs	3 epochs	4 epochs	Baseline (best)	Baseline (4 eps)
snli	64.2%	65.6%	65.1%	65.3%	88.3%	88.3%
mnli	50.3%	51.6%	53.2%	52.8%	69.7%	69.4%
hans_ent	95.8%	96.2%	97.7%	95.8%	99.9%	99.6%
hans_non-ent	10.7%	8.4%	4.5%	8.6%	0.4%	0.4%
hans_n-e (n-e = neut.)	0%	0%	0%	0%	0.04%	0.02%
custom	41.7%	44.4%	50.0%	50.0%	52.8%	52.8%
hypot_alone (emp. str.)	36.9%	37.1%	37.7%	37.5%	49.4%	49.4%

Table 10: Results for Removal of Bias by Flipping Premises and Hypothesis in Contradiction Examples experiment across 4 epochs (model: Enhanced 3). The highest accuracy values across epochs are highlighted in pink and the lowest in blue. The baseline values are given for the reference on the right: they are the best result values for the baseline model across all 4 epochs (Baseline best) and the baseline model's performance after the 4th epoch (Baseline 4 eps). The baseline model is ELECTRA trained on SNLI (see Table 9).

Evaluation Data	Model (2 eps of training total)	Baseline (BART after 2 eps)
snli	74.5%	93.0%
mnli	71.6%	84.7%
hans_entailment	89.5%	99.0%
hans_non-entailment	46.2%	36.0%
hans_non-ent (non-ent = neutral)	6.6%	1.0%
hans_all	67.9%	67.5%
custom	77.8%	75.0%
hypot_alone (single sent.)	53.5%	47.6%

Table 11: Results for Removal of Bias by Flipping Premises and Hypothesis in Contradiction Examples experiment (model: Enhanced 4). The baseline values are given for the reference on the right: they are the the baseline model's performance after 2 epochs of training (BART trained on SNLI, see Table 6).

alone to a degree. Enhanced 4 did not lose this ability, likely because it was trained for 1 epoch in the same way as the baseline model.

Additionally, Enhanced 4 performs slightly better than its baseline model on the custom dataset, and Enhanced 3 performs slightly worse. However, the changes in performance on the custom dataset are too small to draw any conclusions.

In the other evaluations (on SNLI, MNLI, and on HANS entailment examples), Enhanced 3 and Enhanced 4 do not perform better than the baseline models, and in most cases they perform substantially worse. We have three possible explanations as to why this might be happening.

First, it is possible that the models lose some of the undesired bias coming from *contradiction* examples in the dataset, but they pick up additional bias that might be coming from the premises instead. The bias coming from the premises might be worse, which is why the performance is worse overall.

Another possible reason for the decrease in per-

formance is that there might be trickier examples in the dataset that we missed. In those examples, the flipping of the premises and hypothesis might be compromising the correctness of the label. Enhanced 3 and Enhanced 4 might be exhibiting the drop in performance because they might have been trained, or partially trained, on the incorrectly labeled data.

The third explanation for the decrease in performance on SNLI and MNLI validation sets is that both of these datasets are carrying similar bias due to the creation strategy of those datasets. This bias is present not only in the training data, but in the validation data as well. When we attempt to debias the models by flipping the premises and hypotheses, it stands to reason that the models would perform worse on the datasets where the bias is still present (such as validation sets of SNLI and MNLI).

To summarize, Enhanced 3 and Enhanced 4 exhibit some evidence that some of the undesirable bias could have been removed. However, the overall performance on the benchmark datasets SNLI and MNLI degrades significantly.

5 Exploration of DRiFt algorithm to remove bias (models: Enhanced 5 and Enhanced 6)

This part of the project is a reproduction effort with additional experimentation.

Debias by Residual Fitting (DRiFt) algorithm was introduced by He et.al. (2019), and in the present work we conduct two major experiments aimed to remove the Hypothesis Alone bias and the Entailment bias using this algorithm. The resulting models are Enhanced 5 and Enhanced 6.

5.1 Implementation Details

The baseline model for these two experiments is BERT trained on MNLI data. The accuracy values are given in Table 12. The best results across all 4 epochs are taken as the baseline values for Enhanced 5 and Enhanced 6 models.

Enhanced 5 model was obtained by training BERT on MNLI data using the DRiFt algorithm for 4 epochs. The hyperparameters are the same as for the baseline model.

The DRiFt algorithm requires a biased model during the training. The biased model for Enhanced 5 was obtained by training BERT model on MNLI data for 3 epochs, with data modification to create Hypothesis Alone setting. More specifically, during the training, the premises in the input were masked out, so the model could only see and make its prediction based on the hypothesis. After the training, the biased model achieved 60.9% accuracy on MNLI "matched" validation set if the input presented to the model was in the same format as during the training (i.e., if the premises were masked out). When the input was in the standard format (the premises not masked out), the biased model achieved 57.7% accuracy.

To train Enhanced 5 model, the way the loss was calculated has been modified according to the DRiFt algorithm. The specifics of the change are the following:

During the training, Enhanced 5 would produce an output for a specific input (a batch). Subsequently, the input would be changed to the masked format required by the biased model. The masked input would be fed to the biased model, and the biased model would produce its own output. Both outputs would be added together, and the loss would be calculated using torch.nn.CrossEntropyLoss. During the optimization step, the changes were propagated to Enhanced 5 model's weights.

The results for Enhanced 5 model are presented in Table 13.

The experiment to obtain Enhanced 6 model was designed in a similar way, however, we wanted to target the Entailment bias instead of the Hypothesis Alone bias. In order to do so, the biased model was trained differently, and there were some modifications to the way the loss was calculated as well.

The biased model for Enhanced 6 experiment was ELECTRA model trained on MNLI data for 1 epoch, with data modification. Instead of seeing the actual data, the model could see only three types of strings: "Sequence" (if the hypothesis sentence is a subsequence of the premise sentence), "Words" (if all words in the hypothesis sentence belonged to the set of words from the premise sentence), and "Blank" (all other cases). The resulting biased model could not perform well on any of the datasets. However, it predicted *entailment* when presented with "Sequence" or "Words" strings as an input.

When training Enhanced 6, the input was changed for the biased model accordingly, so that the biased model could see the strings "Sequence",

Evaluation Data	1 epoch	2 epochs	3 epochs	4 epochs
mnli	82.3%	83.5%	84.1%	83.8%
snli	77.2%	77.5%	78.9%	79.1%
hans_entailment	99.3%	99.4%	99.0%	98.5%
hans_non-entailment	2.9%	2.3%	4.2%	7.5%
hans_non-ent (non-ent = neutral)	0%	0%	0%	0.04%
custom	47.2%	58.3%	61.1%	61.1%
hypot_alone (attention mask)	37.7%	40.0%	38.0%	36.9%

Table 12: Performance of BERT model trained on MNLI data across 4 epochs (baseline model). The highest accuracy values across epochs are highlighted in pink and the lowest in blue. Values highlighted in color are taken to be the baseline accuracy values for DRiFt experiments.

Evaluation Data	1 epoch	2 epochs	3 epochs	4 epochs	Baseline (best result)
mnli	76.0%	76.7%	78.8%	78.5%	84.1%
snli	71.6%	71.4%	73.6%	73.6%	79.1%
hans_entailment	97.0%	94.6%	96.3%	91.9%	99.4%
hans_non-entailment	5.9%	11.7%	10.5%	18.1%	7.5%
hans_non-ent (non-ent = neutral)	0.5%	0.8%	1.1%	2.0%	0.04%
custom	47.2%	52.8%	58.3%	61.1%	61.1%
hypot_alone (attention mask)	31.5%	28.2%	28.9%	28.9%	36.9%

Table 13: Results for Removal of Bias via DRiFt algorithm experiment across 4 epochs (model: Enhanced 5). The highest accuracy values across epochs are highlighted in pink and the lowest in blue. The baseline values are given for the reference on the right: they are the best result values for the baseline model across 4 epochs (see Table 12).

Evaluation Data	1 epoch	2 epochs	3 epochs	Baseline (best result)
mnli	79.4%	80.5%	80.9%	84.1%
snli	72.6%	73.9%	74.7%	79.1%
hans_entailment	97.6%	98.8%	98.3%	99.4%
hans_non-entailment	5.1%	4.8%	6.4%	7.5%
hans_non-ent (non-ent = neutral)	0%	0%	0.01%	0.04%
custom	38.9%	52.8%	55.6%	61.1%
hypot_alone (attention mask)	38.8%	38.0%	37.4%	36.9%

Table 14: Results for Removal of Bias via DRiFt algorithm experiment across 3 epochs (model: Enhanced 6). The highest accuracy values across epochs are highlighted in pink and the lowest in blue. The baseline values are given for the reference on the right: they are the best result values for the baseline model across 4 epochs (see Table 12).

"Words", or "Blank". For MNLI data, most of the input was turned into "Blank". For those examples, the biased model would not interfere with the training in any way. If the changed input was "Sequence" or "Words", we added the logits from the biased model before calculating the loss. In other words, the main difference between Enhanced 5 and Enhanced 6 methods of training is the fact that the training was changed for all examples for Enhanced 5 model (All Logits method) and only for a very small fraction of examples for Enhanced 6 model (Some Logits method). This was done for Enhanced 6 model because the biased model for that experiment was extremely limited, and we did not want it to interfere with the training on most of the dataset.

The results for Enhanced 6 model are presented in Table 14.

Additional experimentation was conducted without the subsequent analysis. More specifically, we experimented with All Logits versus Some Logits methods for targeting Hypothesis Alone bias. We also tried training for more epochs, tried training on SNLI dataset (instead of MNLI), and tried to get as close as possible to the implementation details reported by He (2019).

Lastly, we tried to use our own baseline models as the biased models for the experiments. In this case, we modified the input before feeding it to the biased model to concur with the Hypothesis Alone setting. Due to time limitations, the comprehensive evaluation of the results was not conducted, and the results are not reported in the present work.

5.2 Results and Discussion (Enhanced 5 and Enhanced 6 models)

Enhanced 5 model performs better on HANS non-entailment examples. The reasons are discussed in Section 3.

Enhanced 5 also shows evidence of the deterioration of the model's ability to make a prediction based on hypothesis alone. This is seen in the results for the performance in Hypothesis Alone setting. Likely, we observe this result because the Hypothesis Alone bias was targeted in a direct way for Enhanced 5.

Enhanced 6 model did not show improvement on any of the datasets. We hypothesize that this is likely because of the deficiency of the biased model: the biased model was not given enough information to make a useful prediction, and it can potentially produce only three outputs, which is very limiting. For a better result, it should consider the percentages of the word overlap in a premise-hypothesis pair instead of whether the word overlap occurred or it did not.

6 Conclusion

We tried three methods to remove bias and obtained some evidence that at least some of the bias could have been removed.

Most of the evidence comes from the performance of the models on non-entailment examples of HANS dataset. However, since there is a labeling discrepancy in how HANS dataset is labeled versus what the models are trained to predict, we are not convinced that the models became better (more robust or better reasoning).

We argue that in the strictest sense, all *non-entailment* sentences in HANS dataset should be labeled as *neutral*. Originally, we did label them as *neutral* and calculated the accuracy results based on that (presented in the tables as *"hans_non-ent (non-ent = neutral)"* measurement). We saw some improvement for Enhanced 1, Enhanced 4, and Enhanced 5 models, but the accuracy values are still relatively low.

Most of the improvement on HANS non-entailment examples arises because of the following: for *non-entailment* examples, the models tend to switch from predicting *entailment* to predicting *contradiction*. While we posit that it is a better prediction, the exact value of this switch remains unclear due to the ambiguity of labeling for HANS *non-entailment* examples.

We have not seen substantial improvement on the custom dataset in any of the enhanced models. It is surprising given that most of our experiments targeted the removal of the Hypothesis Alone bias, and the custom dataset was designed to catch the presence of that bias. However, the custom dataset is too small, and a larger dataset would have been preferable for these experiments.

We have not seen the improvement in the performance on the corresponding out-of-distribution datasets, and in some cases the enhanced model performed worse.

Additionally, we also ran quick evaluations on other datasets such as GLUE Diagnostic Dataset (Wang et al., 2018) and Stress Tests (Naik et al., 2018). We did not include those evaluations in this report because they were not conducted in a comprehensive manner. However, the superficial analysis shows that the enhanced models did not improve compared to the baseline models.

If we were to conduct more experiments, we would like to use the DRiFt method while targeting the Entailment bias specifically. The experiment would be similar to Enhanced 6, however, we would affect all examples during the training instead of a small fraction. As a biased model, we would use a simple model that makes a prediction based on the word overlap between the premise and the hypothesis as well as the length of the sentences. For evaluation, we would use HANS dataset and the "word_overlap" split of Stress Tests. If there was an improvement in performance on HANS as well as on Stress Tests, or if we saw an improvement on any other datasets, then we would be convinced that the DRiFt algorithm was successful for our experiment.

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. *CoRR*, abs/1508.05326.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and

Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. *CoRR*, abs/2003.10555.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Matt Gardner, Yoav Artzi, Victoria Basmova, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hanna Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. 2020. Evaluating NLP models via contrast sets. *CoRR*, abs/2004.02709.

He He, Sheng Zha, and Haohan Wang. 2019. Unlearn dataset bias in natural language inference by fitting the residual. *CoRR*, abs/1908.10763.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. *CoRR*, abs/1707.07328.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101.

R. Thomas McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. *CoRR*, abs/1902.01007.

Aakanksha Naik, Abhilasha Ravichander, Norman M. Sadeh, Carolyn P. Rosé, and Graham Neubig. 2018. Stress test evaluation for natural language inference. *CoRR*, abs/1806.00692.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. *CoRR*, abs/1805.01042.

Antonio Torralba and Alexei A. Efros. 2011. Unbiased look at dataset bias. In *CVPR 2011*, pages 1521–1528.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for NLP. *CoRR*, abs/1908.07125.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. *CoRR*, abs/1804.07461.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *CoRR*, abs/1704.05426.