A study of methods to address MNLI data set artifacts in the Electra-Small LLM

Jose Rojas
University of Texas - Austin University
jlrojas@utexas.edu

Abstract

Models trained on benchmark data sets can be prone to over-fitting on invalid data correlations that are less common in generalized settings. These data set artifacts, such as entailment bias, can be uncovered and counteracted. I examine this bias in the MNLI data set using the Electra-Small large language model. The analysis reveals that the model's entailment bias is linked to correlations of similarity between data set features, which in turn is used identify a specific case of data set examples that are commonly misclassified, subject/object swaps. I evaluate three techniques for addressing this problem case and show how effective they are at reducing its misclassification. The results prove successful for addressing the specific problem case while improving the model's generalization to out-of-distribution data, albeit at the cost of some in-distribution performance.

1 Introduction

Natural language processing research has achieved recent state of the art performance using large language models (LLM). To assist with NLP research of LLMs, benchmark data sets are provided for training and testing. In this paper, a model known as the Electra model (Clark et al. [2020]) will be trained and analyzed using the MNLI benchmark data set to understand its behavior and how the model under-performs on certain tasks. Techniques to improve the failure cases will be evaluated experimentally and analyzed.

2 Model and Data set Artifacts

2.1 Baseline Model Introduction

The model used for the baseline analysis is the Electra-small model, fine-tuned on the MNLI data set (Wang et al. [2018]). The MNLI data set contains two primary features, *premise* and *hypothesis*, and a label for each sample that is used to classify whether the hypothesis *entails*, *contradicts*, or does neither, denoted as *neutral*. This classification task requires a model that interprets the semantic meaning of the words in sentence pairs and uses the given information and logic to infer how the two statements can relate to each other, as task referred to as "Natural Language Inference" ("NLI"). The baseline model is trained for 4 epochs, after which the model showed signs of little further improvement. The accuracy of the baseline model achieves 87% accuracy on the MNLI training set and 82% accuracy on the MNLI validation set (2).

2.2 Model Characteristics

Firstly it is useful to explore the model's behavior given some elementary aspects of data set as a foundation for any biases that may exist. The training and validation classification results have a slight bias towards under predicting *neutral* compared to the other two classes 1. This bias will be examined more closely later.

		% of predictions	
Label	Acc. (%)	Baseline	Hypo-Only
Entail.	82.1	33.3	32.5
Neut.	79.4	32.8	29.4
Contr.	84.8	33.8	38.1

Table 1: MNLI baseline label accuracy & prediction distributions

Data Set	Accuracy (%)
MNLI val.	82.1
HANS	53.5
HANS-1k	0.0
Augmented val.	19.0

Table 2: Baseline Model MNLI Performance, 400k examples

Another behavioral pattern exists between the length of the premise and hypothesis as a function of the accuracy. When we compare the lengths of these two features, premises are commonly much longer than hypotheses. As the premise size grows, the error rate tends to increase, whereas if the hypothesis grows, the error tends to stay the same or decrease (1). This behavior suggests that the model performs well when the premises and hypothesis are a similar length and if there is an imbalance in the premise length, this will lead the model astray from a clearer choice in classification. Also, this may be signal the model may be using a hypothesis as the guiding feature in classifications (Poliak et al. [2018]).

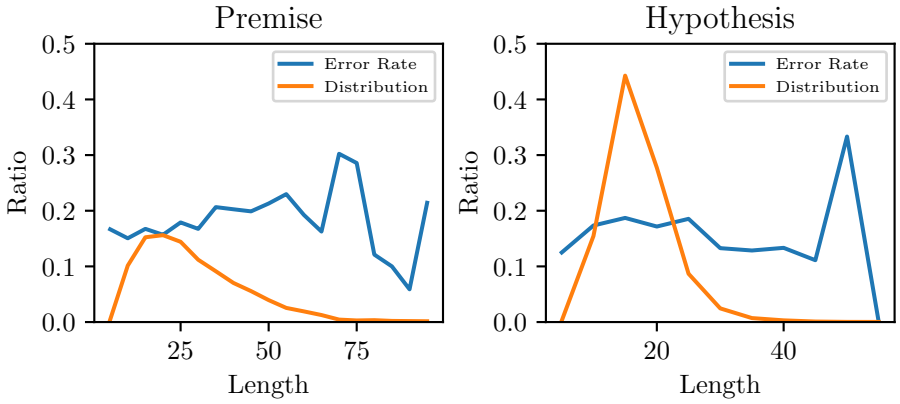

Figure 1: Premise/Hypothesis Classification Characteristics (Error rate vs Length) for Baseline MNLI model.

2.3 Model Ablations

To test the hypothesis bias theory, we can compare the effect of only providing a hypothesis to the model as an input [Poliak et al. [2018]] (refer to 1 - Hypo-Only). We would expect a model to randomly chose with a probability of 33% for a label given no premise to guide its decision. However, the result of this ablation study shows *contradiction* has a significant 4.8% absolute bias over randomness when hypothesis-only features are used. This suggests there are some underlying correlations in the hypothesis feature which the model is using as a classification signal, likely the presence of negation words. Given a statement and no other input, the model should classify the pair as *neutral*, as there is neither entailment or contradiction. The bias against assigning *neutral* when there is no obvious relationship labeling will be analyzed further making use of this study.

2.4 Contrast Set Analysis

Further analysis of the model's poor performance can be accomplished by dissecting specific examples of predictions and formulating new faulty behavior. I used simply worded examples and modified the sentences to produce premise/hypothesis pairs with shifted labels from the original examples (contrast sets, Gardner et al. [2020]). The model overall exhibited some resiliency to these contrast

Premise	Hypothesis	Label Gold	Pred.	Possible Reason
'We are unprohibited'	'We are disallowed'	C	E	Uncommon negation words.
'Fish is not allowed here.'	'The person is allowed.'	N	C	Contr. bias w/ 'not'
'The lawyer saw the judge.'	'The judge saw the lawyer.'	N	E	High similarity entail. bias
'Bother's house after Jacob's apartment'	'Jacob's apartment after Bother's house'	C	E	High similarity entail. bias

Table 3: Examples that perform poorly in baseline model.

C = Contr., E = Entail., N = Neut.

edits, however some general types of failures were discovered. For example, the model fails to properly predict the label when the subject and object are reversed in a way that changes the original semantics: *"The lawyer scolded the judge"*/*"The judge scolded the lawyer"* should be *neutral* but predicted as *entailment*. This can be generalized as a case for any pair that uses the form "SUBJECT VERB OBJECT" and reversing the subject and object phrases without changing other words, referred to as "subject/object swaps". This word swap usually results in a *neutral* gold label as swapping subject/object usually creates a semantic shift which the two statements cannot logically be proved/disproved (except for mutual verbs like "married"). In this example, the judge and lawyer could both have seen or not seen one another, thus providing no clear entailment or contradiction.

Additionally, there are other examples of general failures that perform poorly that were discovered using these probing techniques: temporal relationships (i.e. 'before' and 'after' is not well understood), spatial relationships (i.e. inverting 'from X to Y'/'from Y to X' are incorrectly labeled as entailment), negations other than not (i.e. using uncommon words with 'un-'/'dis-'/'mis-' prefix usually yields a misunderstanding of entailment/contradiction). A table of examples is provided (3).

2.5 Out-of-distribution Analysis

By testing the model on another data set, we can analyze the generalization of the model. I chose one particular out-of-distribution ("OOD") data set for comparison. HANS (McCoy et al. [2019]) is a diagnostic data set used to uncover frailties in models by offering contrast examples not often incorporated into benchmarks. The table shows how poorly this model does on generalization compared to baseline MNLI (2). HANS is of interest in particular as it offers a subset of samples with heuristic-based patterns. The subject/object swap is one particular case included and it is of interest because the model does extremely poorly, achieving 0% accuracy on these examples. This smaller subset on HANS will be referred to as "*HANS-1k*".

3 Model Experimentation & Improvements

I've chosen three methods for attempting to address the biases and artifacts in the Electra model. To reduce the amount of testing effort, I chose the subject/object swap artifact as the general class of errors to address as it is straight-forward to understand, has a low threshold for performance improvement, and requires the least amount of edits to existing data samples. The three methods I've chosen to address this problem are fine-tuning using forgettable sampling, data augmentation, and adding syntax labeling as a sub-task to improve the model architecture.

3.1 Fine-tuning using Forgettable Sampling

Based on the code provided by Swayamdipta et al. [2020], I investigated which examples in the data set were *soft* or *hard* "forgettables", examples which were incorrect at least once during the training process Yaghoobzadeh et al. [2019]. The degree of variability or "forgetfullness" for these examples is correlated to how difficult the example is to learn and how close it may be to a decision boundary for classification. Examples that have high label variability ("soft") are likelier to be learned correctly than an example that has never had the correct label during training ("hard").

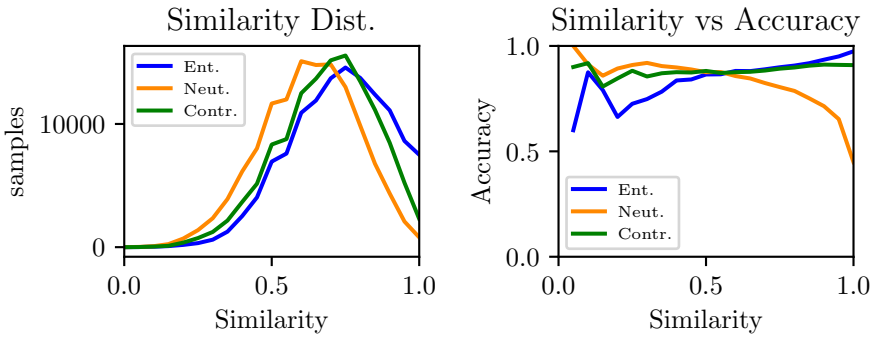

Figure 2: Premise/Hypothesis Classification Characteristics for Baseline MNLI model.

Another method to classify forgettable examples is to utilize the biases discovered with class and premise/hypothesis similarity. To identify potentially weakly classified examples, I used the amount of overlapping words as a similarity measure between the premise and hypothesis statements. I then plotted the distribution of similarity measures against the accuracy per class label (2). From the similarity distribution plots, a class imbalance (Guo et al. [2008]) exists were neutral labels with high similarity measure are poorly classified. This aligned well with subject/object swap class as they tend to yield a neutral label (2.4). Combining these two facts together, a set of examples can be isolated for improvement with a high similarity filter. The reasoning is that subject/object swaps share the same set of words, thus have a similarity measure, and tend toward neutral gold labels. Given the lower accuracy for high similarity neutrals, this cohort of examples can be targeted for potential improvement by focusing further training on samples with similar characteristics. Fortunately the HANS-1k is such a validation set; these samples have high similarity and are all labeled neutrally. This both HANS as a whole and this smaller specific OOD subset are used as a OOD challenge benchmark for generalization for these fine-tuned models.

For the training, I created various subsets of data to fine-tune against. Based on the recommendations in Utama et al. [2020] & Yaghoobzadeh et al. [2019], I trained a weak-model using hypothesis only forgettables as this model could be used as a de-biasing baseline while offering a larger variety of samples on which the Electra model could generalize. I also used the subset of examples trained on the BOW model presented in Yaghoobzadeh et al. [2019] as the highest performing subset that could improve OOD HANS accuracy. Additionally I also choose a subset of random mistakes and high similarity mistakes to test my similarity hypothesis. Using these subsets, I fine-tuned the models for at least 3 additional epochs until there was no discernible improvements and evaluated the accuracy against the OOD validation sets (4).

3.2 Data Augmentation

Within the MNLI data set, there are few to no observable examples of the subject/object inversion (McCoy et al. [2019]). Given the dearth of samples, the model will likely not identify the semantic difference with swapping the subject and object in a sentence that would create a change in inference. To remedy this, adding more examples that exhibit this inversion should help the model recognize this difference. For this set of experiments, I generated a contrast challenge set of 90 examples, sourced by modifying samples from within MNLI. I chose to utilize the original in-distribution data set as the source material so as to avoid having the word distribution or grammatical style of my language from influencing the results. The new examples have about 50% neutral labels, and 25% each of entailment and contradiction labels. I created a train/validation split of about 75% and 25% each with balanced label distributions across both. However, the label imbalance is difficult to avoid for this type of contrast example; a subject/object swap usually generates a neutral semantic meaning as discussed (2.4). I then fine-tuned the network with these examples in addition to a larger cohort of at least 5000 examples or more to prevent the model from over-fitting to the tiny amount of augmented data. To increase the influence of the examples, I added sample re-weighting to the loss function and applied a 10x weight ratio to these augmented samples, without which the training would take much

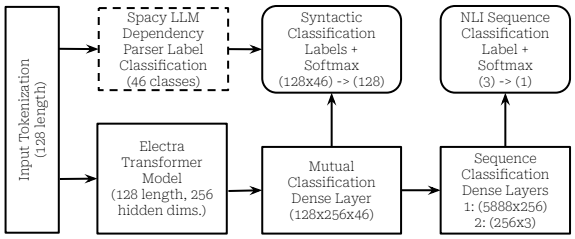

Figure 3: Custom Multitask Model with both Syntax inference and NLI Sequence classification.

longer to see an effect. From the experiment results (4), this challenge set is difficult for the baseline model as it scores less than 20% accuracy on the validation examples.

3.3 Multitask Syntactic and NLI Classification

The previous two approaches changed the training data, but how could the model architecture be changed to fix the subject/object swap problem? Subject/object inversion is a change in semantics based on a shift in syntactic structure between words, so one possibility is to train the model to better identify syntactic elements of the entire sentence [Sachan et al. [2020]]. To achieve this goal, I generated the syntactic labels for the token classification task by using the Spacy dependency tree parser which uses a LLM to infer dependency tree tags for tokens in the input features. The original model was extended with a linear layer to classify each input token's hidden state from a total of 46 dependency labels. I then added the sequence classification layer after these layers. I experimented with a few different arrangements of the classification layers, which all did well at each individual task, but with no substantial improvement on sequence classification. However the essential structural choice made to enable the best sequence classification for this model is by forcing the sequence classification to classify with syntactic knowledge. This knowledge is gained by training each token with syntactic labels and by only linking the sequence classification layer to a mutually shared linearly layer used by syntactic token classification. A diagram of the model architecture is shown (3).

Training this revised model requires generating the Spacy token labels, which is a slow process, increasing the model training time. Caching of the processed data set was required to improve the performance of these experiments. The experiments included training a full multitask model and comparing its performance on MNLI, HANS, fine-tuned performance on forgettables, and performance with data augmentation. Given time constraints on training time, only a few avenues were tested.

Generally what was observed is this model has strong syntactic classification abilities, with over 95% accuracy on all labels. Also, there was rapid convergence on syntactic token classification, reaching the 90+% accuracy within a fraction of an epoch. This implies the pre-trained Electra transformer model has built-in syntactic knowledge and only minor fine-tuning was needed.

4 Result Analysis

4.1 Which models worked the best at resolving the chosen artifacts?

The results of the various model experiments demonstrate definitive performance gains. There is a clear performance gain on all OOD data sets by fine-tuning on a specific subset of training examples, however the degree of performance varies. Overall the two most high performing models are the forgettable sampling derived from the BOW models and its counterpart that I generated, the forgettable sampling with re-scaling of the weak hypothesis only model. These both achieved increased performance on the HANS and HANS-1k sets, showing a boost in classification for the subject/object inversion case (98.9% and 40.7% respectively over the 0% of the baseline) while retaining high in-distribution (ID) performance (75.5% and 80.1% vs 82.1%).

Models that incorporated data augmentation or high-similarity de-biasing achieved increased performance on the augmentation challenge set. High similarity forgettables examples of the baseline

Model	Accuracy (%)				Train set size
	MNLI val.	HANS	HANS-1k	Aug. val.	
Baseline (B)	82.1	53.5	0.0	19.0	400k
Hypothesis Only (HO)	57.3	47.3	8.2	38.1	400k
Multitask w/ Syntax (MT)	81.2	53.1	0.0	19.0	400k
(B) + Aug.	31.8	50.0	*100*	*57.1*	70
(B) + Rand(HNeg(B))	62.6	60.9	3.5	28.5	9k
(B) + Rand(HNeg(B)) & Aug.	55.7	50.6	*100.0*	*52.3*	9k
(B) + Neg(HiSim(B))	42.6	49.9	*100.0*	*57.1*	9k
(B) + Neg(HiSim(B)) & Aug.	42.6	50	*100.0*	*61.9*	9k
(B) + Neg(HO)	81.5	56.5	4.1	19.0	90k
(B) + Neg($HO_{rescaled}$)	**80.1**	**62.9**	**40.7**	19.0	90k
(B) + Neg($HO_{rescaled}$) & Aug.	**80.4**	**59.9**	**99.0**	19.0	90k
(B) + Neg(BOW)	75.7	**59.0**	**98.9**	19.0	64k
(B) + Neg(BOW) & Aug.	75.2	**55.3**	**99.8**	**28.6**	64k
(MT) + Neg($HO_{rescaled}$)	80.1	53.6	3.5	19.0	90k
(MT) + Neg($HO_{rescaled}$) & Aug.	**79.6**	**58.8**	**40.2**	**33.3**	90k
(MT) + Neg(BOW) & Aug.	66.6	**54.0**	**95.5**	**52.3**	64k

Table 4: Empirical results from model training experiments on MNLI data set.

Legend: '+' = fine-tuning. Neg = forgettable sampling. HNeg = Hard forgettable sampling. Rand = a random selection, HiSim = High similarity filter, BOW = BOW forgettable sampling set from Yaghoobzadeh et al. [2019], $_{rescaled}$ - Similarity-Label rescaling., Aug. = Augmented data set. **Bold** = note-worthy successful results, *Italic* = over-fitted successful results

model boosted performance (57.1%, 61.9% with augmentation vs 19%) while the syntactic multitask model with augmentation also did well on this task (52.3%).

Comparing these results against a model that uses random sampling of forgettables for fine-tuning, the random model's performance is inconsistent - it does slightly better on the augmented validation set (close to chance of 33%) but does poorly on HANS-1k (only 3.5%) while reducing its ID performance considerably.

Overall the model fine-tuned on forgettable re-scaled sampling did the best at classifying subject/object inversions in HANS-1k (40.7% vs 0% on the baseline) and maintaining high generalization accuracy on MNLI and HANS.

4.2 What effect does fine-tuning on a smaller subset of data have on model performance?

Based on the empirical accuracy results, it is clear we can achieve better performance on OOD data sets with focused training on smaller batches of representative samples. Fine-tuning on the BOW weak model's forgettables offered the most balanced and consistent results as it improved HANS by 6% in total accuracy from the baseline. It also boosted the subject/object inversion on HANS-1k by a staggering 98% while also improving on the adversarial augmentation data set by 9%. To understand why, the similarity plots are helpful. The plots for the BOW forgettable samples shown illustrates this model is weaker on a different distribution of the data as a function of the similarity measure (4). The number of examples in the upper range of the similarity measure results in a better balance of class labels compared to the entire training set (2). Fine-tuning on this distribution creates an opportunity for the minority class *neutral* labels to improve their performance as they are more fairly represented (5). The result is a sharp improvement in HANS and HANS-1k performance which additionally improving on augmentation data set. This is consequence of the subject/object swap examples having a high similarity measure (2.4).

Utilizing the weak hypothesis-only model's forgettables slightly improved HANS/HANS-1k performance (by 3% and 4%). To further test the similarity-bias theory, in one experiment (4 see 'Rescaled'), I re-weighted the same forgettable examples by their class labels across the similarity distribution. For example, high similarity examples were re-weighted so the minority neutral samples had equal representation compared to the more dominant entailment samples, while lower similarity examples

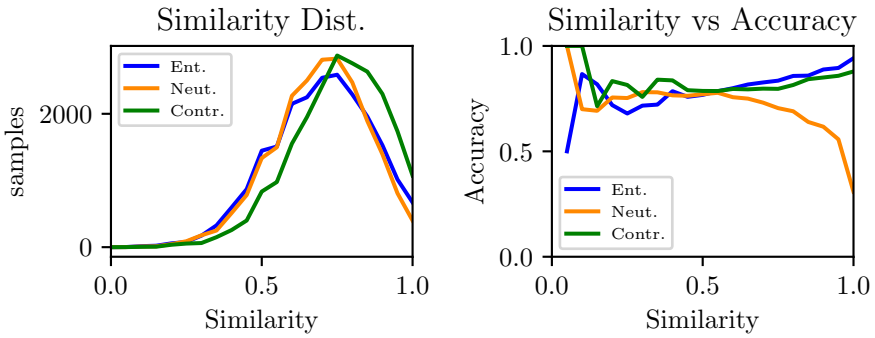

Figure 4: Similarity/Accuracy plot of BOW forgettable sampled fine-tune training set
Note: the distribution of labels in the high-similarity range is more balanced compared to baseline plot 2. As well, the neutral labels are far more likely to be incorrect, allowing for a greater margin of improvement in the model's challenge set performance.

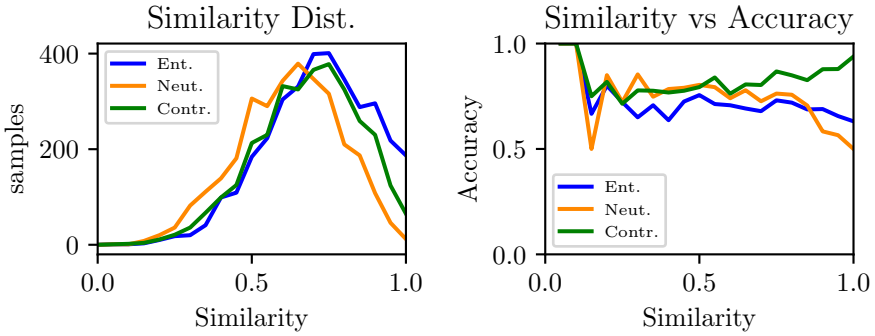

Figure 5: Similarity/Accuracy plot of BOW forgettable fine-tuned model on MNLI validation set
Note: entailment bias for high-similarity samples has been removed, accuracy results for neutral are improved

had the entailment samples weighted higher. This changed boosted model performance, achieving the best OOD accuracy on HANS with a 9% improvement and additionally a 40.7% improvement on HANS-1k, beating the baseline and many other fine-tuning results. This model achieved also one of the highest in-distribution performance measures, beating the BOW model by 5%.

4.3 Does fine-tuning on a smaller subset always work well?

Fine-tuning performance is greatly affected by the characteristics of the subset of examples. Further evidence can be seen in the experimental results. By reducing the fine-tuned training set size, a radical shift can be made to improve OOD performance. In two sets of trials, I fine-tuned only on the forgettables with high similarity and on a random set of mistakes of the original baseline model. In the random case of 9k samples, HANS, HANS-1k, and augmented accuracy increased by 7%, 3.5% and 9.5% whereas on high similarity forgettables decreased HANS by 3.5% but increased the HANS-1k and Augmented performance by 100% and 57.1%, yielding the best performance boosts on the subject-object inversion case. I believe this demonstrates how similarity-entailment bias in the MNLI data set can be targeted directly by re-sampling in a more representative distribution that eliminates this bias. However such re-sampling has its side-effects; both of these subsets performed extremely poorly with in-distribution accuracy, both declining over 20% in the MNLI validation compared to the baseline. This shift is mostly a function of the number of samples, as the BOW and HO-Rescaled models used many more samples (64k and 90k) compared to these smaller subsets, offering more variety and achieving more balanced performance improvements across ID and OOD

data sets. The take-away is that fine-tuning should contain a large amount of widely representative samples of the generalization that is sought, while balancing and counteracting any specific biases in the original data set.

4.4 How well does data augmentation work?

Data augmentation has mixed performance in the experiments that it was utilized. Fine-tuning directly with the augmented training set alone boosted performance at the expense of the MNLI validation accuracy (a decrease to 31.8%), as the fine-tuning will over-fit to the augmented data. A better practice would be to add the data to a cohort of larger samples for fine-tuning. However, it would be crucial to re-weight the loss function for the augmented samples by a significantly large factor to see any discernible affect.

In some cases where the augmentation was added to additional samples, the augmentation did not work in any predictable way. This is likely caused by a low scaling factor or biases in the accompanying training samples with counteracting the influence of the augmentation samples.

What can be surmised is there are challenges to overcome with augmentations: (1) the augmented data must be re-weighted to increase its influence in the training set, (2) it may not be included with other data that counteract the effects of the augmented data, (3) the augmented validation set should match the characteristics of the augmented training examples, otherwise the training can over-fit without seeing an increase in validation accuracy.

The best strategy with augmentations would be to ensure the augmentation set is relatively large to maximize the training influence and append it to a robust and varied set of examples for fine-tuning to avoid over-fitting.

4.5 Does adjusting the model architecture have any benefits?

The syntactic multitask model did very well on syntactic labeling task (reaching near perfect results 97+%) but did not increase OOD performance on the NLI task alone, achieving only similar performance to the baseline model. This is expected: the model fits the data, and the data has biases that under-represent certain cases in the generalized setting. Changing the model architecture only helped increase performance to other existing models fine-tuned on de-biased subsets, for example, the Multitask model boosted augmented challenge set performance to 52% from 28.6% with the same fine-tuning data set parameters which was one of the best results. However it did not boost performance in the forgettable re-scaled data subset, which was not as strong in subject/object inversions as the BOW model. This suggests the model can help improve convergence by finding short-cuts in the gradient path; it is likely the BOW model without the multitask architecture could have increased performance independently, although more slowly. The takeaway is that model architecture will assist with convergence but will not assist with generalization in poorly biased data sets without a redistribution of the training examples.

5 Conclusions

This study investigated the bias artifacts in the MNLI data set. An entailment bias as a function of the similarity between the premise and hypothesis statements was uncovered. This leads to entailment over-fitting for cases when the pair of statements have similar words. A general case of failure affected by the entailment bias, the subject/object swap, was analyzed and tested across various models. Three model changes were analyzed: fine-tuning on forgettables, data augmentation, and a multitask model approach using syntactic labeling. All three methods showed various degrees of improvement on OOD data sets and challenge sets, however the most reliable method was fine-tuning on forgettable/forgotten samples of weak models, particularly BOW and Hypothesis-Only based models. The overall performance improved across the targeted OOD data sets and challenge set, however the side-effect in all cases was reduced in-distribution performance. By addressing the model data set biases through improving a particular failure case, inevitably, the model's baseline performance will be reduced as it will no longer overfit to the original data set characteristics. By exposing these biases and their effects, it can be concluded that models which train on benchmark data sets with artifacts can have 'blind-spots' and can over-estimate their generalization performance.

References

K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*, 2020. URL https://openreview.net/pdf?id=r1xMH1BtvB.

M. Gardner, Y. Artzi, V. Basmova, J. Berant, B. Bogin, S. Chen, P. Dasigi, D. Dua, Y. Elazar, A. Gottumukkala, N. Gupta, H. Hajishirzi, G. Ilharco, D. Khashabi, K. Lin, J. Liu, N. F. Liu, P. Mulcaire, Q. Ning, S. Singh, N. A. Smith, S. Subramanian, R. Tsarfaty, E. Wallace, A. Zhang, and B. Zhou. Evaluating models' local decision boundaries via contrast sets, 2020. URL https://arxiv.org/abs/2004.02709.

X. Guo, Y. Yin, C. Dong, G. Yang, and G. Zhou. On the class imbalance problem. *Fourth International Conference on Natural Computation, ICNC '08*, Vol. 4, 10 2008. doi: 10.1109/ICNC.2008.871.

T. McCoy, E. Pavlick, and T. 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*, pages 3428–3448, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1334. URL https://aclanthology.org/P19-1334.

A. Poliak, J. Naradowsky, A. Haldar, R. Rudinger, and B. Van Durme. Hypothesis only baselines in natural language inference, 2018. URL https://arxiv.org/abs/1805.01042.

D. S. Sachan, Y. Zhang, P. Qi, and W. Hamilton. Do syntax trees help pre-trained transformers extract information?, 2020. URL https://arxiv.org/abs/2008.09084.

S. Swayamdipta, R. Schwartz, N. Lourie, Y. Wang, H. Hajishirzi, N. A. Smith, and Y. Choi. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of EMNLP*, 2020. URL https://arxiv.org/abs/2009.10795.

P. A. Utama, N. S. Moosavi, and I. Gurevych. Mind the trade-off: Debiasing nlu models without degrading the in-distribution performance, 2020. URL https://arxiv.org/abs/2005.00315.

A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding, 2018. URL https://arxiv.org/abs/1804.07461.

Y. Yaghoobzadeh, S. Mehri, R. Tachet, T. J. Hazen, and A. Sordoni. Increasing robustness to spurious correlations using forgettable examples, 2019. URL https://arxiv.org/abs/1911.03861.