Digging up Dataset Artifacts in SNLI

Eric Culbertson
University of Texas at Austin
ericmculbertson@utexas.edu

Brian Ho
University of Texas at Austin
brianho@utexas.edu

Abstract

Although significant advances have been made in the NLP space in recent years, model generalization continues to be hindered by dataset artifacts, correlations within data that models learn rather than the underlying language tasks. This paper explores the presence of such artifacts in the SNLI dataset, specifically when learned by the ELECTRA-small model. Through methodologies such as CheckList and adversarial challenge sets, we uncover unique and abnormal behaviors of the model and propose techniques to correct them.

1 Introduction

1.1 Dataset Artifacts

Those in the machine learning space are constantly aiming for the coveted achievement of strong generalization: the sign that your model performs well not only on your training set but also on real-world data it has never seen before. This is no different for NLP. One threat to generalization that is commonly discussed in the NLP space is the concept of dataset artifacts, spurious correlations in a dataset that do not necessarily correspond to the underlying task being studied. And as explainability for more complex models continues to be elusive, models trained on datasets containing artifacts could very well just be learning these correlations rather than actual patterns rooted in natural language. As a result, these models would struggle when tested on examples that go against these correlations, even those very similar to the data it was trained on. Conversely, they could perform unreasonably well on broken sets that do align with the correlations but would otherwise be impossible for a human to solve.

1.2 Natural Language Inference and SNLI

In this paper, we seek to analyze and mitigate dataset artifacts within the context of the ELECTRA-small model trained on the Stanford Natural Language Inference (SNLI) dataset (Clark et al., 2020; Bowman et al., 2015). The task the dataset serves to teach is Natural Language Inference, the ability to discern whether a given hypothesis is an entailment, a contradiction, or neutral with respect to a given premise.

Prior research has identified that SNLI contains strong correlations between the hypotheses on their own and the labels. In fact, models trained on the dataset can even approach 70% accuracy on the accompanying validation and test splits when given only hypotheses (Poliak et al., 2018). This varies immensely from the 33% by random chance that one would expect if NLI was truly being performed. In an effort to explain this peculiarity, a study by Gururangan et al., 2018 found that SNLI contains a few specific biases that will cause models to associate certain characteristics of hypotheses with labels. In particular, hypotheses containing negation are frequently labeled as contradiction, while those containing generic or approximate words are commonly labeled as entailment. But one need not think hard to come up with real-world examples that defy these patterns.

Other investigations have revealed SNLI-trained models to lean on shallow syntactic relationships between premises and hypotheses to predict labels. For example, if the two share words, contiguous subsequences, or parse subtrees, the model will tend to predict entailment (McCoy et al., 2019). Clearly, there exist many fortuitous patterns within SNLI that systems can exploit to achieve high performance without needing to confront the semantic complexities of NLI. Our study strives to expand on the discoveries made in this area and contribute insights that can help train more robust and versatile models.

2 Baseline Analysis

ELECTRA-small trained on the training split of SNLI with 3 epochs achieves 89.6% accuracy on the validation split. This strong performance can be misleading though, and the following analysis section will utilize various techniques to highlight problem areas within SNLI that could potentially hamper the model's performance on real-world examples.

2.1 CheckList

For a given dataset, for example SNLI, the standard practice is to use train-validation-test splits, where the held out data is used to estimate the performance of the model at a given task. Over time, many models are developed that perform incrementally better on these held out sets. The optimistic end goal in these cases is to meet or exceed human performance on this held out set, where performance is estimated as a single number representing accuracy, F1, or some other metric. As shown in other research, this can lead to problems when both the training set and validation set have unmeasured biases, and therefore performance on real world data is overestimated (Rajpurkar et al., 2018; Patel et al., 2008; Recht et al., 2019).

Addressing the shortcomings that are inherent with only evaluating the model on a predefined test set, a testing paradigm and Python library called CheckList was developed. This testing paradigm combines many evaluation methods proposed in prior literature, such as robustness to noise (Belinkov and Bisk, 2018), adversarial changes (Ribeiro et al., 2018), fairness (Prabhakaran et al., 2019) and logical consistency (Ribeiro et al., 2019). The CheckList framework allows for model evaluation to go beyond a single performance metric, towards a more comprehensive model evaluation paradigm. This provides researchers with easy to construct methods to see if the model is truly learning language structure rather than spurious trends in the dataset.

Following the methodology of the CheckList paper, the trained ELECTRA-small model was evaluated against a series of tests meant to evaluate various specific capabilities of the model. These capabilities tested included *NER* (ability to recognize and understand named entities), *Fairness* (whether the model clearly had unintended biases towards gender or race), *Negation*, *Logic* (symmetry, consistency, conjunctions, disjunctions, etc), and *Robustness*.

NER

In order to test the model's ability at NER, the first set of tests developed were designed to see if the model could correctly identify the importance of names. For example, if the premise is "Robin is staying at home." and the hypothesis is "Julia is at home", this should be a contradiction since different named entities are in the premise and hypothesis. Using the checklist library, examples of "same action different entity" pairs are generated using the CheckList Template object. As shown in table 1, the model fails to distinguish different named people 36.7% of the time. Named entities were quite rare in the training set, so this performance is expected. We'd likely see similar performance at distinguishing named locations.

Fairness

In order to test fairness, a test was constructed that perturbs the original validation set by swapping gendered words in a sentence to construct an analogous premise and hypothesis that should have the same prediction. Ideally, the gender of the subjects in the hypothesis should have no effect on the truth of that hypothesis. For example, (**Premise:** "The man is in medical school", **Hypothesis:** "the man wants to be a doctor") should have the same predicted value as (**Premise:** "The woman is in medical school", **Hypothesis:** "the woman wants to be a doctor"). As shown in table 1, the "change gender" test has a failure rate of 2.0%, indicating that the model predicts similarly regardless of gender.

Taxonomy

Ideally, a model that is well-performing in NLI should have an understanding of synonyms and antonyms. So a series of tests were developed to test that capability. As an example, (**Premise:** "He is trying to be more careful", **Hypothesis:** "He is trying to become less reckless") should be entailment, while (**Premise:** "He is trying to be more careful", **Hypothesis:** "He is trying to become less deliberate") should be a contradiction. Pairs of synonyms and antonyms were generated using a BERT model provided in CheckList's Template object. As shown in Table 1, the model performs perfectly at identifying that "wants to be more/less `{synonym}`" is a contradiction, but fails 100% of the time to identify that "wants to be more/less `{antonym}`" is entailment. In all

Capability	Test Name	Failure %	Example Failure
NER	different people	36.7	(p: Jack is back at NBC, h: Mary is back at NBC) → ent. (cont.)
Fairness	change gender	2.0	(p1: A boy walking in the park, h1: A young man is in the park) ≠ (p2: A girl walking in the park, h2: A young woman is in the park)
Taxonomy	more/less synonym	0.0	
	more/less antonym	100.0	(p: Mary is more religious, h: Mary is less secular) → cont (ent)
	most/least synonym	3.2	(p: Brian is the most humble, h: Brian is the least modest) → ent. (cont.)
	most/least antonym	98.8	(p: Ray is the most fat, h: Ray is the least thin) → cont. (ent.)
Robustness	add one typo	19.8	(p1: A man dances., h1: A man is dancing.) ≠ (p2: A man dances., h2: A mna is dancing.)
	contractions	0.8	(p1: A man purchases food., h1: He is cooking a big meal.) ≠ (p1: A man purchases food., h1: He's cooking a big meal.)
Logic	both A and B ent.	0.0	
	both A and B cont.	0.0	
	neither nor ent.	25.2	(p: The woman likes to neither surf nor hike, h: She doesn't like to surf) → cont. (ent.)
	neither nor cont.	100.0	(p: The woman likes to neither relax nor reflect, h: She likes to relax) → ent. (cont.)

Table 1: CheckList Tests Failure Rates for baseline model.

cases the model is predicting contradiction, which gives evidence that the model is focusing on the words "more" and "less" in the hypothesis and prediction, and hasn't truly learned synonyms and antonyms. This capability was tested again using superlatives, in the tests "is the most/least `{synonym}` in `{mask}`" and "is the most/least `{antonym}` in `{mask}`". Performance is similar, where the model predicts almost every example as a contradiction.

Robustness

Tests were also made to evaluate the model's robustness to random typos, and contraction modifications ("Don't" → "do not"). The model was found to be exceptionally robust to contractions, meaning that there is evidence that the model understands equivalence between "don't" and "do not", but had a higher failure rate to typos. Robustness to typos is important when an NLP model is evaluated over text that is not professionally edited, but less important for other tasks. This should be kept in mind when deploying a model trained on finely curated input data.

2.2 Custom adversarial challenge sets

Another technique frequently used to discover problems in NLP models is to test them on adversarial challenge sets, which are intended to trick systems without being overly complicated or misleading humans. In one notable test, we took the default validation split of SNLI and repeated each hypothesis. When tested on this set, accuracy drops significantly from 89.6% to 49.6%. It is noteworthy that this accuracy calculation assumes that the true labels of the set do not change after the modification. Even though this assumption may not be completely valid given the possibility for altered interpretation of some hypotheses after duplication, what is even more unusual than the accuracy is the distribution of the model's predicted labels. While the model has a very uniform distribution of predictions over the original dataset, it predicts over 83% of the examples in the repeated set as entailment. Many examples such as the following were correctly labeled as contradiction in the original but switched to entailment after the alteration.

> Premise: Two men on bicycles competing in a race.
>
> Hypothesis: A few people are catching fish. → **Contradiction**
>
> Hypothesis: A few people are catching fish. A few people are catching fish. → **Entailment**

To investigate this phenomenon further, all hypotheses in the set were then changed to the same nonsensical sentence: "The pizza goes to Jupiter on a rocket ship." As expected, the model predicts everything as a contradiction. The hypotheses were then nearly repeated, with only the final word missing: "The pizza goes to Jupiter on a rocket ship. The pizza goes to Jupiter on a rocket." The model still labels almost every example as a contradiction. However, with the inclusion of the last word, the model suddenly predicts 95% entailment. These findings suggest that through training on SNLI, the model somehow learned to associate hypotheses composed of two repeating sections with the entailment label, regardless of their content or corresponding premises. One possible explanation is that the examples in the training set with the highest ratio of hypothesis to premise length are all entailments. But this does not explain the drastic change in prediction distribution from the addition of the final word in the duplicated nonsense set.

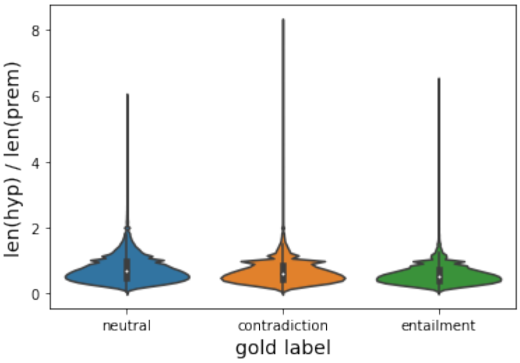

Figure 1: Distributions of hypothesis/premise length over the different labels.

3 Methods

Different techniques were evaluated to assess their value in addressing the shortcomings identified.

3.1 Training on adversarial data

Research by Jia and Liang, 2017 recommends training on adversarial examples to mitigate biases due to dataset artifacts in models. Zhou and Bansal, 2020 expand on this with what they call 'Data-Level Debiasing,' but caution that the technique has limits in its effectiveness and is disadvantageous due to requiring users to know the specific biases in advance. But for the purposes of our investigation, it proved to be quite effective.

To deal with the model's tendency to predict examples with repeated hypotheses as entailments, we took the first 1000 examples of SNLI's training split and duplicated their hypotheses. The already trained model was then trained on these additional examples. Although this is no doubt an extremely simple implementation, the abnormal sensitivity of the model's behavior with this particular artifact suggests that even a small presence of debiasing measures during training should have a noticeable impact.

3.2 Data augmentation

Models are often limited by the information available in the dataset that is used for training. Often a model's training set does not have an appropriate amount of variability, or has some inherent biases that limits the model's ability to generalize. Aggregating more data from different data sources and manually annotating that data can be time consuming and infeasible. This infeasibility motivates the practice of data augmentation, where labeled training data can be altered in a plausible way in order to increase the variety of information that the model can use to learn from. This technique was first popularized in computer vision, but has seen wide adoption in NLP as well (Feng et al., 2021).

There are a variety of potential augmentation techniques that can make sense for the application of NLI. These augmentations can be either targeted, where specific types of sentences are augmented in order to fix known issues, or general. The four general data augmentation techniques used were backtranslation, synonym replacement (Zhang et al., 2015), contextual insertion, and contextual augmentation (Kobayashi, 2018). Backtranslation used pretrained language translation models to translate from english, to french, back to english in order to add natural plausible variation. Synonym replacement used Word2Vec embeddings to replace random words with another

Dataset	**Original Model**	**Fixed Model**
	Acc (% E / N / C)	Acc (% E / N / C)
SNLI validation	89.6 (34 / 33 / 33)	87.7 (33 / 33 / 34)
Repeated SNLI validation	49.6 (83 / 3 / 14)	88.1 (34 / 33 / 33)
Nonsense	100.0 (0 / 0 / 100)	100.0 (0 / 0 / 100)
Nearly repeated nonsense	94.9 (5 / 0 / 95)	100.0 (0 / 0 / 100)
Repeated nonsense	0.1 (100 / 0 / 0)	100.0 (0 / 0 / 100)

Table 2: Accuracy and distribution of predicted labels of the original and fixed models on the various datasets.

word with similar embeddings. Contextual insertion used a pretrained BERT model to randomly add a word that is plausible given the context of the sentence. Finally, contextual replacement picked a random word that appeared in either the hypothesis or premises, and swapped all occurrences of that word with a plausible replacement.

In contrast to the general augmentations mentioned above, targeted augmentations can create data specifically to address known deficiencies identified by CheckList. For example, since the model fails on the "more/less antonym" examples, these sentence structures can be upsampled specifically to fix the failing capability. While targeted augmentations like these are cheaper to generate and more directly address known model limitations, they are limited because testers likely are unable to explicate all model deficiencies. Therefore a model trained using the targeted augmentations could be less likely to generalize. For this reason, only general augmentations were considered. 15,000 random augmented examples were generated from the SNLI training set, and these examples were used to fine tune the trained ELECTRA-small model.

4 Results

The results of training on the additional examples with duplicated hypotheses are detailed in Table 2. The model's accuracy on the original SNLI validation set drops by roughly 2%, but its performance on the repeated set is vastly improved, with accuracy jumping from 49% to 88%. The distribution of its predicted labels across the repeated set also moved from strongly skewed towards entailment to very uniform. On all 3 nonsense sets, the revamped model labels every example as contradiction. The small training adjustment was therefore able to completely alleviate the model's irrational behavior on our custom datasets without sacrificing much performance on the original validation set.

As an additional measure of generalization, we tested the original and improved models on the various rounds of the ANLI dataset from Nie et al., 2020 (Table 3). This is a challenging NLI set derived from Wikipedia passages that was constructed through human annotation with a model-in-the-loop component. The improved model actually performs slightly better than the original in rounds 2 and 3, despite it being unlikely that the set would feature any examples with a repeated hypothesis. This reveals that the improvement may have also slightly bolstered the model's ability to handle longer examples in general.

Separately, the performance of the ELECTRA-small model fine-tuned using data augmentation was also tested. The performance of the augmentation model was identical on the SNLI validation set, but as shown in table 3, the model trained with augmentation showed consistent improvement on the ANLI challenge sets. This provides evidence that augmentation successfully added some useful variability not present in the SNLI training set. However, the performance on the various checklist tests listed in table 1 was uniformly unchanged or slightly worse. It is unlikely that these failing CheckList cases could be resolved with the augmentation techniques used, since the augmentations are more effective at adding vocabulary variety than variety in sentence structure. The exception to this is the typo capability test, where augmentations that added random typos in the training set would likely have a positive effect on model robustness to typos. This was not tested however. More testing would also be needed to see which augmentation methods were the most beneficial to increasing generalizability to the ANLI challenge sets.

Training method	**A1**	**A2**	**A3**
Standard	0.304	0.304	0.317
Repeated	0.304	0.316	0.322
Augmented	0.305	0.320	0.329

Table 3: Performance on ANLI test sets.

5 Conclusion

Should we expect models to approach human capability on the Natural Language Inference task, likely cultivated through decades of experience with the language, it has become increasingly apparent that training on enormous datasets can still leave numerous problems on the table. As demonstrated in our study, even models trained on a renowned set like SNLI can have their flaws, like being unable to navigate examples containing antonyms coupled with “more”/“less” or completely veering off course when given repeated hypotheses. Fortunately, even simple modifications can result in strong improvements in these areas. As researchers in the NLP field continue to dissect the interactions between models and data, one can be optimistic that great progress will be made to reduce the negative impacts of dataset artifacts and illuminate the way towards more powerful systems.

References

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180-191. Association for Computational Linguistics.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. 2019. Do imagenet classifiers generalize to imagenet?. In *International Conference on Machine Learning*, pages 5389–5400.

Kayur Patel, James Fogarty, James A Landay, and Beverly Harrison. 2008. Investigating statistical machine learning as a tool for software development. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 667–676. ACM.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pretraining Text Encoders as Discriminators Rather Than Generators. In *Proceedings of the International Conference on Learning Representations (ICLR)*

Marco Tulio Ribeiro, Carlos Guestrin, and Sameer Singh 2019. Are red roses red? evaluating consistency of question-answering models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics.*

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin 2018. Semantically equivalent adversarial rules for debugging nlp models. In *Association for Computational Linguistics (ACL).*

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don’t know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784-789, Melbourne, Australia. Association for Computational Linguistics.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021-2031, Copenhagen, Denmark. Association for Computational Linguistics.

Rongzhi Zhang, Yue Yu, and Chao Zhang. 2020. SeqMix: Augmenting Active Sequence Labeling via Sequence Mixup. In *PIn Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8566–8579. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Em- pirical Methods in Natural Language Processing*, pages 632–642. Association for Computational Linguistics.

Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *n Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, Association for Computational Linguistics.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, Eduard Hovy 2021. Survey of Data Augmentation Approaches for NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021 Processing (EMNLP)*, Association for Computational Linguistics.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R. Bowman, 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. Association for Computational Linguistics.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. 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. Association for Computational Linguistics.

Vinodkumar Prabhakaran, Ben Hutchinson, and Margaret Mitchell. 2019. Perturbation sensitivity analysis to detect unintended model 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) Linguistics (Volume 2: Short Papers)*, pages 5740–5745, Hong Kong, China. Association for Computational Linguistics.

Xiang Zhou and Mohit Bansal. 2020. Towards robustifying NLI models against lexical dataset biases. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8759–8771, Association for Computational Linguistics.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Association for Computational Linguistics.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *International Conference on Learning Representations.*