The roots and effects of heuristics in natural language inference and question answering models

Jungyhyun Min*
Department of Cognitive Science,
Johns Hopkins University

R. Tom McCoy**
Department of Cognitive Science,
Johns Hopkins University

Tal Linzen†
Department of Linguistics and
Center for Data Science,
New York University

The advent of the PAID paradigm, where pre-trained language models (often transformer-based) are fine-tuned to create "high performance" models, has accelerated progress in natural language processing. Leaderboards on classification tasks like natural inference (NLI) and question answering (QA) are populated with scores nearing or even surpassing human baselines. However, studies suggest this paradigm is faulty and does not indicate humanlike natural language understanding: the popular methodology of fine-tuning on training set and evaluating on test set from the same distribution may not accurately measure the robustness of a model. Unlike humans, fine-tuned models adopt statistical heuristics even in high-resource environments and show unstable and poor out-of-distribution generalization, despite high test set performance. In this study, we track down heuristics in various stages of training of BERT-based NLI and QA models to study how heuristics arise, how they affect out-of-distribution generalization, and how they may be mitigated. We conclude that heuristics arise from pre-trained representations and insufficient fine-tuning, and they may transfer across tasks. These heuristics can be effectively mitigated by syntactic data augmentation. Throughout the paper, we propose immediate methodologies to mitigate drawbacks of the PAID paradigm.

1. Introduction

Recent advances in machine learning, including the transformer and pre-trained language models, has produced a plethora of models, each claiming to outperform the previous.

Since the turn of the century, countless artificial intelligence technologies, many driven by machine learning models, have joined the human's ordinary life. Machine translation, facial recognition, and automated vehicles are some examples where machine learning models drive everyday technologies. In both academic literature and the press, there are discussions on how quickly neural network-based artificial models are

* E-mail: jmin10@jhu.edu.
** E-mail: tom.mccoy@jhu.edu.
† E-mail: linzen@nyu.edu

improving. There are a number of ways in producing and evaluating neural models, and the Pretraining-Agnostic Identically Distributed (PAID) paradigm (Linzen 2020) is one of them.

1.1 The PAID paradigm and NLI

Linzen (2020) describes the PAID paradigm in three parts. First, a word prediction model is pre-trained on a corpus. Second, a the model learns a target task by fine-tuning on a training set, which comprise of examples representing the task. Finally, the model is evaluated on a test set that comes from the same distribution as the training set. Such paradigm is prevalent in the field of natural language processing, often championed by transformer-based models (Devlin et al. 2019; Raffel et al. 2020) and their variants trained for classification tasks listed on the GLUE leaderboard (Wang et al. 2018). The paradigm has garnered extraordinary growth in the field, closing on or even surpassing untrained human baselines in some tasks (Linzen 2020; Wang et al. 2019).

One task where models using the paradigm excel is natural language inference (NLI), also referred to as recognizing textual entailment (RTE). NLI is the task of identifying inference. A premise and a hypothesis are provided, and a successful NLI model is able to identify whether the premise entails, contradicts, or is neutral to the hypothesis. NLI models fine-tuned and evaluated on the multi-genre NLI corpus (Williams, Nangia, and Bowman 2018) report remarkable performance on the MNLI test set, with the T5 model surpassing GLUE's human baseline (Raffel et al. 2020).

However, the PAID paradigm is not without limits.

1.2 Limits of the PAID paradigm

Limits of the PAID paradigm include its popular yet insufficient evaluation methodology, which uses data from the same distribution as the model's training dataset to evaluate a model's performance. This in-distribution only evaluation is insufficient because brittle heuristics may exist in datasets. Statistical learners like neural networks may learn to wrongly connect a distribution-specific heuristic to a linguistic feature. Examples of such heuristics include syntactic or word patterns visible to the human brain, like correlation of a specific word to a certain label, and those only recognized by computational processors like correlation of the inverse log of sequence probability to labels. While these heuristics may be helpful in solving a specific distribution of data, they may not generalize to a different dataset representing the task.

McCoy, Min, and Linzen (2019) show that in NLI, high accuracy on a test set does not necessarily entail that the model has mastered the task, and that "good" models may sometimes get the right answers not by understanding the sentences, but rather by relying on shallow heuristics, which are consistent with the majority of test set examples but fail with more complex and difficult examples. Similar behaviors persist in other areas of natural language understanding and in computer vision (Agrawal, Batra, and Parikh 2016; Jia and Liang 2017; Wang et al. 2017; Weber, Shekhar, and Balasubramanian 2018).

1.3 Questions we attempt to answer

In this paper we explore these limits of the PAID paradigm and potential remedies that may be effective to a statistical learner like BERT (Devlin et al. 2019) by asking four questions. One, where do the heursitics discussed in McCoy, Pavlick, and Linzen (2019)

originate, and how they are learned? Two, what is a possible method help improve out-of-distribution performance by reducing reliance on the heuristics? Three, how do hyperparameters like seed and training set order affect BERT's behavior with adopting heuristics? And four, how specific are our findings to the MNLI dataset, or the NLI task?

2. Heuristics in BERT-MNLI models

2.1 Background: heuristics ML models, MNLI and otherwise

Machine learning models often score well on test sets by adopting heuristics that are effective for frequent examples. However, while effective for frequent examples, such heuristics often cause the seemingly successful model to fail in more complex cases.

We analyze this topic of heuristics with respect to one popular task: natural language inference (NLI), also known as recognizing textual entailment (RTE). The task, first formally discussed by the likes of Montague (2008) and Cooper et al. (1996), involves determining whether a given premise entails, or infers the truth of, the premise. NLI datasets, including RTE-1 (), SciTail (), and SNLI (Bowman et al. 2015) thus comprise of premises, hypotheses, and entailment labels.

(1) a. **Premise:** The lawyer by the actor ran.
b. **Hypothesis:** The actor ran.
c. **Label:** Non-entailment.

(2) a. **Premise:** The author read the book.
b. **Hypothesis:** The author read.
c. **Label:** Entailment.

In this paper, we discuss the Multi-genre NLI dataset (Williams, Nangia, and Bowman 2018), or MNLI, which is an extension on the SNLI corpus. A human-annotated corpus for natural language inference, MNLI was compiled by crowdworkers who were presented premises from 12 genres and were asked to generated entailing or non-entailing hypotheses.

Research has found there are several properties of the MNLI dataset that could influence low-bias learners to adopt shallow heuristics. To a certain degree, labels can be predicted by looking at hypotheses only (Poliak et al. 2018; Gururangan et al. 2018), and some words show high correlation with certain labels (Mu and Andreas 2020). In particular, negation words are aligned with non-enatilment labels (Kim et al. 2019).

By design, statistical learners are incentivized to adopt heuristics and take advantage of such artifacts and patterns in the MNLI dataset, so that they result in high test set performance. The lexical overlap heuristic (McCoy, Pavlick, and Linzen 2019), where the NLI system assumes entailment when there is lexical overlap between the premise and the hypothesis, is no accident; we later show that lexical overlap is correlated with entailment in MNLI by plotting label counts across premise-hypothesis overlap rates. Such distribution likely originates from human heuristics employed by crowdworkers when they are asked to generate hypotheses that entail a given premise. To complete the task with minimal time and effort, crowdworkers may employ repetitive strategies resulting in high correlation between certain words or properties such as negation words and word overlap and labels. However, there are also correlations by chance. It is unlikely that words like "sleeping" were prompted in a crowdworker's attempt to expedite task completion. Correlation between a non-entailment label and the word

Heuristic	Definition	Example
Lexical overlap	Assume that a premise entails all hypotheses constructed from words in the premise	**The doctor** was **paid** by **the actor**. $\nrightarrow$ [WRONG] The doctor paid the actor.
Subsequence	Assume that a premise entails all of its contiguous subsequences.	The doctor near **the actor danced**. $\nrightarrow$ [WRONG] The actor danced.
Constituent	Assume that a premise entails all complete subtrees in its parse tree.	If **the artist slept**, the actor ran. $\nrightarrow$ [WRONG] The artist slept.

Table 1
The heuristics targeted by the HANS dataset, along with examples of incorrect entailment predictions that these heuristics would lead to. (Figure from McCoy, Pavlick, and Linzen (2019).)

"sleeping" (Mu and Andreas 2020) is probably due to an accidental high frequency of the word in examples with the non-entailment label.

But do models actually capture these statistical properties in MNLI? We first note that models are able to achieve high scores on the MNLI dataset. For instance, BERT achieves state-of-the-art performance after only 3 epochs of fine-tuning (Devlin et al. 2019). In addition, using MNLI in addition pre-training boots performance in semantic similarity (Reymers and Gurevych, 2019) and Boolean QA (Clark et al. 2019). But this high performance alone is consistent with using the heuristics, or with not using them. Recent works find that deep learning models learn simple heuristics that are "ecologically valid" in training set (Dasgupta et al. 2018) and high performing models often fail simple sanity checks (Naik et al. 2018). McCoy, Pavlick, and Linzen (2019) propose HANS as out-of-distribution evaluation set, which targets three heuristics: the lexical overlap, subsequence, and constituent heuristics, defined in Table 1, and is able to analyze their adoption by models. McCoy, Pavlick, and Linzen (2019) got an initial characterization of models' performance on HANS. In order to get a better grasp of the nature and extent of the problem, we conduct a more thorough analysis of BERT's handling of HANS.

2.2 Experiment: BERTs of a feather (McCoy, Min, and Linzen 2019)

We train 100 BERT-base MNLI models, with varying classifier initialization and order of the training set. Other hyper-parameters are consistent across all 100 runs, and follow that from Devlin et al. (2019). Each model consisted of BERT, and a linear classifier on top. Across the 100 instances, all instances of BERT were initialized identically with weights from bert-base-undcased, while the instances of linear classifier were initialized with random weights, each instance different from the next. During the finetuning stage (3 epochs), weights of both BERT and classifier were updated. We also varied the order of MultiNLI training examples for each model. We use the framework established in McCoy, Pavlick, and Linzen (2019), in which models are trained on MNLI training set, evaluated in-distribution generalization on MNLI development set, and evaluated out-of-distribution generalization on HANS McCoy, Pavlick, and Linzen (2019).

The models showed high and stable performance in in-distribution generalization. All 100 models, regardless of initial classifier weight and training set order, achieved MNLI development set accuracy in the 0.84 - 0.85 range as shown in Figure 1. However, they showed low and unstable performance in out-of-distribution generalization on

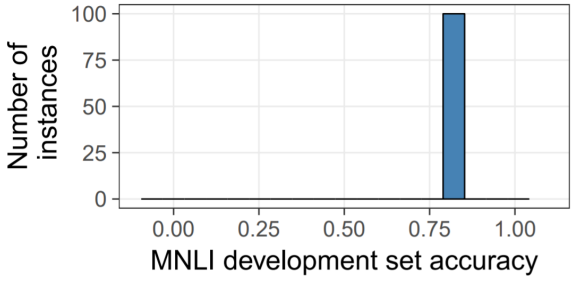


Figure 1
Histogram of MNLI development set accuracy (in-distribution generalization). All 100 models' performance lie within the 0.84 - 0.85 range. Figure from McCoy, Min, and Linzen (2019).

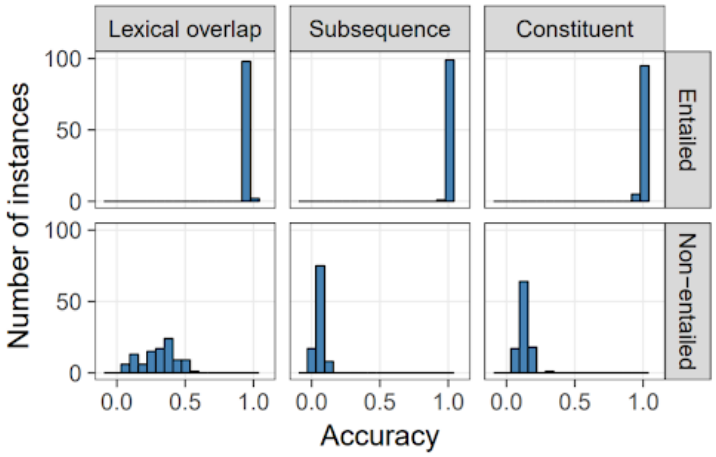


Figure 2
Histogram of HANS accuracy (out-of-distribution generalization), by heuristic and entailment label. While accuracy on entailed examples is consistently high, accuracy on non-entailed examples are low and unstable. they see the most variance in non-entailed examples with lexical overlap. Figure from McCoy, Min, and Linzen (2019).

HANS, as shown in Figure 2. Such significant variance across runs echoes preliminary observations from Devlin et al. (2019). The models' high accuracy on examples whose label is entailment and low accuracy on examples whose label is non-entailment suggest that the models adopt the three heuristics discussed in Section **??**. In addition to this low performance, in the lexical overlap category of HANS, these models showed striking variance, ranging from 0.05 to above 0.50. Such behavior is interesting because the models share everything but the order of the training set and the classifier layer initialization. Further analysis within the lexical overlap show even greater variance within each category of sentences with lexical overlap, as shown in Figure 3. The passive category is an exception, where all models performed near-zero.

We conclude that BERT trained on MNLI adopts the lexical overlap, subsequence, and constituent heuristics, albeit to varying degrees . This echoes behavior previously observed by Devlin et al. (2019) and McCoy, Pavlick, and Linzen (2019), and suggests

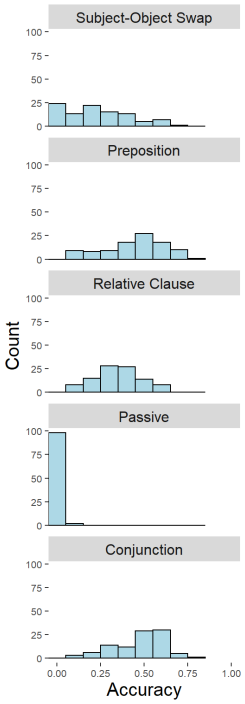
Category	Example	Baseline performance
Subject-object swap	The doctor visited the lawyer. ↛ The lawyer visited the doctor.	
Preposition	The judges by the manager saw the artists. ↛ The artists saw the manager.	
Relative clause	The actors advised the author who the tourists saw. ↛ The author advised the tourists.	
Passive	The senators were recommended by the managers. ↛ The senators recommended the managers.	
Conjunction	The doctors advised the presidents and the tourists. ↛ The presidents advised the tourists.	

Figure 3
Results on the various subcategories within the non-entailed lexical overlap examples of the HANS dataset. Figure adopted from McCoy, Min, and Linzen (2019).

that features like lexical overlap may be linked to the entailment label in the MNLI dataset. To investigate the dataset causes the lexical overlap heuristic, we conduct a simple analysis of rates of lexical overlap across examples of each labels. While McCoy, Pavlick, and Linzen (2019) offer an analysis on the number of datapoint in MNLI that subscribe to the discussed heuristics, we expand on that study by also considering datapoints that do not fully subscribe too the heuristics but may reinforce them.

2.3 Study: overlap analysis

MNLI is unbalanced in that there are many more examples that support the lexical overlap heuristic than those that counter it (McCoy, Pavlick, and Linzen 2019). However, the thousands of supporting and contradicting cases account for less than one percent of the entire dataset of more than $297k$ examples. We extend the analysis by comparing the number of cases supporting and contradicting the lexical overlap heuristic for examples with incomplete overlap. We define overlap rate o as the ratio of number of words in hypothesis also in premise $n(w_{H \cap P})$ to length of the hypothesis in words $n(w_H)$:

$$o = \frac{n(w_{H \cap P})}{n(w_H)}$$

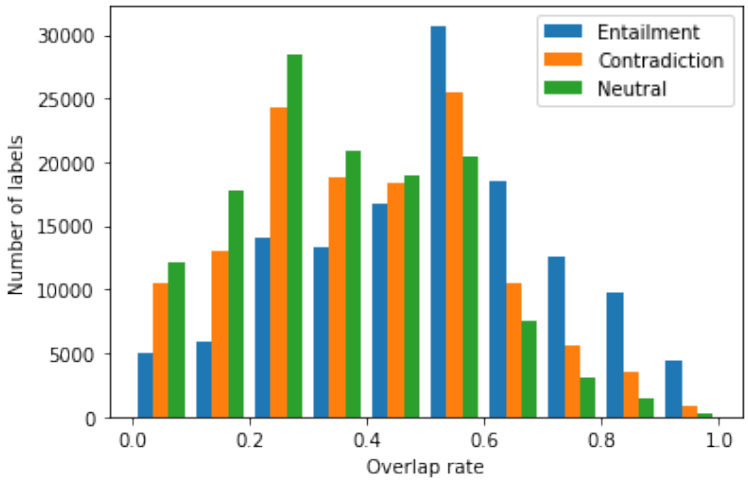


Figure 4
Histogram of MNLI labels by overlap rate.

An MNLI example *There are 16 El Grecos in this small collection* → *This small collection contains 16 El Grecos* has $n(w_H) = 7$ and $n(w_{H \cap P}) = 5$, for overlap rate $o = 0.71$.

A histogram of MNLI gold labels by overlap rate (Figure 4) shows that in the MNLI training set, there are more non-entailing (neutral and contradicting) examples with low lexical overlap, and more entailing examples with high lexical overlap. Such distribution may play a part in influencing models finetuned on MNLI to adopt the lexical overlap heuristic.

2.4 Discussion

According to the results from section 2.2, BERT fine-tuned on MNLI is unstable and vulnerable to heuristics, including the the syntactic heuristics studied in McCoy, Pavlick, and Linzen (2019). The dataset includes imbalances including what we see in **??** and other artifacts that may be responsible for BERT's reliance on heuristics. However, not everything is attributable to the dataset. Inductive bias determined by hyperparameters like seed and model architecture also play a role in producing observed behavior– some instances of BERT-based MNLI models rely less on heuristics than others (McCoy, Min, and Linzen 2019). All 100 instances from McCoy, Pavlick, and Linzen (2019) were presented the same patterns and word frequencies across labels. However, the degree off heuristic adoption varied across instances significantly. Since such heuristics are not likely to be learned by human learners of NLI, instances of BERT that relied more on heuristics May be describes as lacking human-like inductive bias. Human heuristics, accidents, BERT's inductive bias, and BERT's ability to capture statistical properties are all factors in learning heuristics. With regards to the PAID paradigm, this behavior suggests that we need to be more attentive in evaluating model performance. Evaluation on multiple runs of an architecture and using challenge sets like HANS can offer more robust analyses.

3. Data augmentation can be one solution

3.1 Background: improving out-of-distribution generalization

In Section 2, we identify four problems within training under the common implementations of PAID paradigm, two in training, and two in evaluation. In training, models adopt heuristics and do so in an unstable manner. In evaluation, out-of-distribution generalization is often overlooked, and analysis is done on a small number of model samples that does not robustly represent the structure, dataset, and methods used due to the instability. The latter two problems can be overcome with longer fine-tuning (Zhou et al. 2020; Zhang et al. 2020) and analysis over a large number of random restarts (McCoy, Min, and Linzen 2019; McCoy, Frank, and Linzen 2018), but the former two problems require more sophisticated solutions.

Although there are works across computational fields that document instability in generalization (McCoy, Min, and Linzen 2019; Zhou et al. 2020; Wendlandt, Kummerfeld, and Mihalcea 2018; Verma and Zhang 2019), few go on to propose solutions beyond balancing of training datasets (Zhou et al. 2020) and analysis over many reruns (McCoy, Min, and Linzen 2019; McCoy, Frank, and Linzen 2018).

On the other hand, there are many proposed solutions to improving out-of-distribution generalization. Data augmentation can be a simple and effective method to do so, often used in classification tasks such as NLI (Yaghoobzadeh et al. 2019; Moosavi et al. 2020; Cengiz and Yuret 2020). We discuss specific methods of augmentation and additional approaches to the problem in Section 3.5.

3.2 Syntactic data augmentation increases robustness to inference heuristics

In this section, we explore the cause of BERT's poor generalization from Section **??**. This section is a recap of Min et al. (2020), where we search for explanations on BERT's poor-performing and unstable performance by considering two hypotheses to explain how BERT fine-tuned on MNLI relies on heuristics and thus fails on HANS.

The Representational Inadequacy Hypothesis explains that BERT relies on heuristics and fails on HANS because its pretrained representations are missing some necessary syntactic information. The Missed Connection hypothesis explains that while BERT has sufficient syntactic information in its pretraining, it fails on HANS because there are not enough MNLi examples that indicate how syntax should support NLI.

To test these hypotheses, we create a small amount of additional training data that is adversarial to the heuristics that BERT adopts during MNLi finetuning. The Representational Inadequacy Hypothesis predicts that such augmentation will have little effect in improving the models' HANS performance and robustness to inference heuristics because BERT lacks necessary syntactic representation for NLI and will need a much larger training data to learn them. On the other hand, the Missed Connection Hypothesis predicts that the small augmentation will provide the 'missed connection' that BERT needs and will improve BERT's performance on HANS.

We create augmentation data by employing two transformations to simple transitive sentences from MNLI (only hypotheses are transformed–MNLI premises are often longer): inversion and passivization. Inversion swaps the subject and object of a sentence. For example, inversion transforms, the suspect followed the detective to the detective followed the suspect. We note that inversion almost always changes the meaning of the sentence, and changes the gold label of any entailing MNLI example. Passivization turns the sentence into the passive voice. There are two passivization

methods–one preserving the meaning by maintaining subject-object order and the other modifying it by swapping them. The same source sentence can be passivized to the suspect was followed by the detective, which preserves the meaning, or the detective was followed by the suspect, which modifies the meaning.

We also employ two composition (premise-hypothesis pairing) methods: original premise, and transformed hypothesis. In the original premise composition, the original premise and the transformed hypotheses are paired as the augmentation example's premise and hypothesis. If the transformed hypothesis maintains its original meaning (this only happens with passivization), the gold label is the example's original label. If the transformed hypothesis no longer retains its original meaning, the gold label for this augmentation example is non-entailment. In the transformed hypothesis composition, the original hypothesis and the transformed hypothesis are paired as premise and hypothesis. Again, if the transformed hypothesis maintains its original meaning, this augmentation's gold label will be entailment. If the transformed hypothesis now carries a different meaning, they have a non-entailment example. All in all, we are able to generate a total of nine augmentation datasets:

- Inversion (original premise)
- Inversion (transformed hypothesis)
- Passivization (original premise)
- Passivization (transformed hypothesis, positive)
- Passivization (transformed hypothesis, negative)
- Passivization (transformed hypothesis, positive + negative)
- Inversion + Passivization (original premise)
- Inversion + Passivization (transformed hypothesis, negative)
- Inversion + Passivization (transformed hypothesis, positive + negative)

In addition to them, we add a random shuffling condition, whose examples are random shuffles of MNLI premises and hypotheses to test whether a syntactically uninformed method could be effective adversarial examples of the heuristics that focus on lexical overlap rather than word order.

Finally, for every transformation-composition combination, they implement three different sizes of augmentation: small (101 examples), medium (405), and large (1215). Small and medium are not subsets of medium and large, respectively, and combined datasets (inversion + passivization, positive + negative) were not uniformly sampled. All datasets are considerably smaller than the MNLI training set ($297k$). The datasets and code used to generate them are available on a Github repository.

We appended each augmentation datasets to the MNLI training set to create augmented training sets, then fine-tuned BERT on it to create the robust model. Five sets of each robust model was created, using different classifier initializations and training set order each time, following prior work (McCoy, Pavlick, and Linzen 2019). They found that in general:

- The transformed hypothesis composition is more effective than the original premise composition.

- Inversion is more effective than passivization.
- The greater the augmentation size, the greater the improvement. However, there are some exceptions where the medium augmentation is more effective than the large.

Overall, augmentation with inversion with the transformed hypothesis dataset was the best strategy in improving out-of-distribution generalization in NLI, reporting not only generalization within the lexical overlap heuristic but also to other heuristics. Augmentation with inversion + passivization with the transformed hypothesis was also effective, reporting highest HANS overall accuracy. The random shuffling condition dataset was not an effective adversarial augmentation set to the lexical overlap heuristic.

	MNLI			Overlap			Subsequence			Constituent		
	S	*M*	*L*	*S*	*M*	*L*	*S*	*M*	*L*	*S*	*M*	*L*
Original premise												
Inversion	.84	.84	.84	.07	.40	.44	.01	.06	.12	.06	.09	.12
Passivization	.84	.84	.84	.23	.35	.54	.04	.05	.09	.13	.11	.15
Combined	.84	.84	.84	.42	.25	.36	.07	.05	.04	.14	.15	.12
Transformed hypothesis												
Inversion	.84	.84	.84	**.46**	**.71**	**.73**	**.09**	**.25**	.23	**.17**	.23	.18
Passivization	.84	.84	.84	.41	.43	.31	.06	.06	.07	.13	.15	.17
Combined	.84	.84	.84	.32	.64	.71	.06	.13	**.28**	.15	**.26**	**.22**
Pass. (only pos)	.84	.84	.84	.30	.20	.29	.04	.04	.05	.10	.13	.11
Pass. (only neg)	.84	.84	.85	.36	.45	.39	.06	.06	.06	.15	.13	.13
Random shuffling	.84	.84	.84	.26	.19	.35	.05	.05	.06	.15	.14	.14
Unaugmented		.84			.28			.05			.13	

Table 2
Accuracy of models trained using each augmentation strategy when evaluated on HANS examples diagnostic of each of the three heuristics—lexical overlap, subsequence and constituent—for which the correct label is *non-entailment* ($\nrightarrow$). Augmentation set sizes are *S* (101 examples), *M* (405) and *L* (1215). Chance performance is 0.5.

We further analyze the successful augmentation using the inversion with the transformed hypothesis dataset by running 10 more iterations, for a total of 15 per augmentation dataset size. We observe that even augmentation with only a hundred adversarial augmentation is able to discourage the model from relying on heuristics, and that the single-format augmentation is able to activate syntactic sensitivity that results in the models improve their out-of-distribution generalization across sentence structures and heuristics.

We see evidence for both the Missed Connection Hypothesis and the Representational Inadequacy Hypothesis. Only a small number of examples disproving the lexical overlap heuristics by swapping the object and the subject was able to increase accuracy on similarly structured examples dramatically (0.19 unaugmented, 1.00 large). The model was able to generalize this learning to other sentence structures with lexical overlap, improving accuracy on examples with lexical overlap from 0.28 (unaugmented) to 0.73 (large). They saw further generalization to sentences with other heuristics, albeit

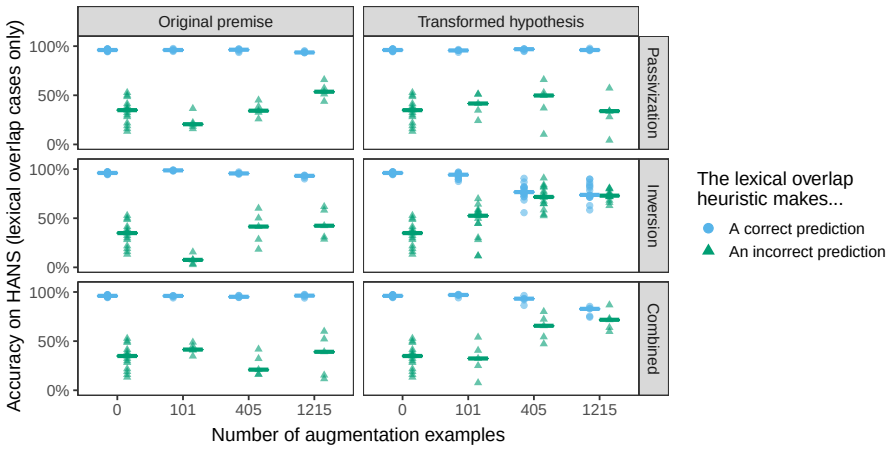


Figure 5
Augmentation using subject/object inversion with a transformed hypothesis. Dots represent the accuracyon HANS examples diagnostic of each of the heuristics, as produced by each of the 15 runs of BERT fine-tunedon MNLI combined with each augmentation data set. Horizontal bars indicate median accuracy across runs.

Subset of HANS	Label	Unaugmented	Small	Medium	Large
MNLI	All	0.84	0.84	0.84	0.84
Subject/object swap	$\nrightarrow$	0.19	0.53	1.00	1.00
All other	$\rightarrow$	0.96	0.93	0.77	0.77
lexical overlap	$\nrightarrow$	0.30	0.44	0.64	0.66
Subsequence	$\rightarrow$	0.99	0.99	0.84	0.85
	$\nrightarrow$	0.05	0.09	0.25	0.23
Constituent	$\rightarrow$	0.99	0.98	0.97	0.97
	$\nrightarrow$	0.13	0.17	0.23	0.18

Table 3
Effect on HANS accuracy of augmentation using subject/object inversion with a transformed hypothesis. Results are shown for BERT fined-tuned on the MNLI training set augmented with the three size of augmentation sets (101, 405 and 1215 examples), as well as for BERT fine-tuned on the unaugmented MNLI training set.

to less prominent degrees. This supports the Missed Connection Hypothesis because a small amount of adversarial argumentation was able to "bridge the gap" between MNLI and the task.

However, we also see that even with hundreds of augmentation with non-entailing passive examples, the model is unable to discard the lexical overlap heuristics with passive sentences (unaugmented 0.01, large 0.01). This suggests that BERT's training lacks sufficient internal syntactic representations to understand passives in the context of NLI, and supports the Representational Inadequacy Hypothesis. We postulate that BERT will need a much larger dataset to learn to better handle passive sentences in NLI.

		Entailment			*Non-entailment*		
Architecture or training method	Overall	*L*	*S*	*C*	*L*	*S*	*C*
Baseline (McCoy, Min, and Linzen 2019)	0.57	0.96	0.99	0.99	0.28	0.05	0.13
Learned-Mixin + H (Clark, Yatskar, and Zettlemoyer 2019)	0.69	0.68	0.84	0.81	0.77	0.45	0.60
DRiFt-HAND (He, Zha, and Wang 2019)	0.66	0.77	0.71	0.76	0.71	0.41	0.61
Product of experts (Karimi Mahabadi, Belinkov, and Henderson 2020)	0.67	0.94	0.96	0.98	0.62	0.19	0.30
HUBERT + (Moradshahi et al. 2019)	0.63	0.96	1.00	0.99	0.70	0.04	0.11
MT-DNN + LF (Pang, Lin, and Smith 2019)	0.61	0.99	0.99	0.94	0.07	0.07	0.13
BiLSTM forgettables (Yaghoobzadeh et al. 2019)	0.74	0.77	0.91	0.93	0.82	0.41	0.61
Ours:							
Inversion (transformed hypothesis), small	0.60	0.93	0.99	0.98	0.46	0.09	0.17
Inversion (transformed hypothesis), medium	0.63	0.77	0.84	0.97	0.71	0.25	0.23
Inversion (transformed hypothesis), large	0.62	0.77	0.85	0.97	0.73	0.23	0.18
Combined (transformed hypothesis), medium	0.65	0.92	0.96	0.98	0.64	0.13	0.26

Table 4
HANS accuracy from various architectures and training methods, broken down by the heuristic that the example is diagnostic of and by its gold label, as well as overall accuracy on HANS. All but MT-DNN + LF use BERT as base model. *L*, *S*, and *C* stand for lexical overlap, subsequence, and constituent heuristics, respectively. Augmentation set sizes are $n = 101$ for small, $n = 405$ for medium, and $n = 1215$ for large.

We conclude that both the Missed Connection Hypothesis and the Representational Inadequacy Hypothesis have merit: while BERT's pretrained representations are in fact missing some necessary syntactic information, there is also a gap between BERT's syntax and NLI that fine-tuning fails to completely bridge. As a result, in addition to this finding, we observe that the low and unstable out-of-distribution performance observed from McCoy, Min, and Linzen (2019) can be somewhat mitigated by syntactic data augmentation, without affecting in-distribution performance (Min et al. 2020).

Syntactic data augmentation, easily generated from a single-sentence source, is effective where there is missed connection between pretraining and the task, but not where there is representational inadequacy within pretraining. It is for now an effective method to mitigate known bias in MNLI.

3.3 Augmentation dataset targeting the subsequence and constituent heuristics

In the previous Section 3.2, we use syntactic data augmentation to increase robustness to inference heuristics. In particular, we augment MNLI training set with counterexamples to the lexical overlap heuristic, created by inverting the subject and object or passivizing. We observe generalization to the other two heuristics–subsequence and constituent–and convincing models to rely less on them. However, non-entailed subsequence is still challenging to NLI models (McCoy and Linzen 2019; McCoy, Min, and Linzen 2019), and we postulate that direct adversarial augmentation to the other heuristics may be more effective at countering the subsequence and constituent heuristics. Thus, we propose two additional augmentation strategies, one targeting the subsequence heuristic by manipulating NP/S sentences, and another targeting the constituent heuristic by manipulating sentences with constituents embedded under a verb.

3.3.1 Subsequence: NP/S. First, we target the subsequence heuristic. The code to generate NP/S augmentation dataset has not been fully implemented. However, we outline the general idea here. In order to find NP/S sentences, we parse through MNLI hypotheses with a verb (V) whose sister is a sentence (S) or a clause introduced by a subordinating conjunction (SBAR). Then, we look at the leftmost child of the S or SBAR. If it is an NP, we can create an augmentation example whose direct object is the NP. If it is not an NP, we keep looking at its leftmost child, the child's leftmost child, and so on until we run into an NP (use the NP as direct object) or a leaf node (we cannot create an NP/S augmentation example).

From an MNLI hypothesis "*Air travel makes a journey by boat unnecessary, but the Suez Canal is still of great importance*," such an algorithm can extract the following transformed hypothesis: "*Air travel makes a journey by boat.*" With this, we can create two types of augmentation datasets, following experiments from Section 3.2. First, the original premise, where the original premise does not entail the transformed hypothesis:

(3) a. **Premise:** Though the advent of air travel negates the need for passengers to take a long sea journey, the Suez Canal is still important for cargo vessels, and watching a leviathan tanker travel sedately through the passage is an unreal experience.
b. **Transformed hypothesis:** Air travel makes a journey by boat.
c. **Label:** Non-entailment

Second, the transformed hypothesis, where the original hypothesis does not entail the transformed hypothesis:

(4) a. **Original hypothesis as premise:** Air travel makes a journey by boat unnecessary, but the Suez Canal is still of great import.
b. **Transformed hypothesis:** Air travel makes a journey by boat.
c. **Label:** Non-entailment

A small modification can also allow another type of subsequence to be generated. Finding a verb (V) whose sister to the left is an S or an SBAR, and looking for an NP as the rightmost descendant of that S or SBAR can generate augmentation examples that are adversarial to the subsequence heuristic:

(5) a. **Premise:** The book on the table is blue.
b. **Hypothesis:** The table is blue.
c. **Label:** Non-entailment

3.3.2 Constituent: embedded under verb. Next, we target the constituent heuristic. To create the embedded under verb augmentation examples, we first parsed through the MNLI corpus for sentences with a verb and a clause under it. Then, we are able to extract the clause. For example:

(6) a. **Premise:** Clark also expressed the hope that he and Redgrave could continue with their marriage.
b. **Hypothesis:** Clark hoped that he could continue their marriage.
c. **Embedded under verb:** He could continue their marriage.

Then, we are able to create two non-entailing augmentation examples by using the clause as the transformed hypothesis

First, we can generate an example in the form of original premise ↛ transformed hypothesis:

(7) a. **Premise:** Clark also expressed the hope that he and Redgrave could continue with their marriage.
b. **Transformed hypothesis:** He could continue their marriage.
c. **Label:** Non-entailment

Second, we can generate an example in the form of original hypothesis ↛ transformed hypothesis:

(8) a. **Origianl hypothesis as premise:** Clark hoped that he could continue their marriage.
b. **Transformed hypothesis:** He could continue their marriage.
c. **Label:** Non-entailment

An important thing to note here is that the label may not seem correct for some augmentation examples. We assume non-entailment for every augmentation example, but if the clause is embedded under a factive verb like prove, know, or confirm, it is also likely that that clause is entailed. However, we did not screen for this, nor did we exclude examples with factive verbs. There are two reasons: one, excluding examples with factive verbs would have reduced the size of our dataset significantly; two, we did not have an effective strategy to parse for sentences with negated factive verbs like They refuse to admit that they are wrong and Over 1100 people have tried to prove that rat choice teachings is wrong. Even though admit and prove are factive verbs, the sentences do not entail that They are wrong or that Rat choice teaching is wrong. As a result, the resulting augmentation dataset is noisy. We name the two datasets constituent with an original premise and constituent with a transformed hypothesis, respectively.

We follow previous work from Section 3.2 and report our augmentation method results in a similar manner. To compare results across different augmentation strategies, we use the same dataset sizes of small (n = 101), medium (n = 405), large (n = 1215), and introduce the extra-large size, where we use all of our generated augmentation data. Because different augmentation strategies generate different numbers of augmentation examples, the extra-large size performances cannot be directly compared across augmentation strategies, and should be thought of as an upper limit of performance improvement using our methodology (modifying training set examples to create augmentation data) MNLI.

3.4 Augmentation results: constituent

In Section 3.3 we created two types of augmentation datasets targeting the constituent heuristic, following the methodology from previous work (Min et al. 2020) described in Section 3.2: embedded under verb with original premsie, and embedded under verb with transformed hypothesis. Both augmentations target the Reflecting their results, we expect constituent with transformed hypothesis to be more effective than constituent with transformed premise.

3.4.1 Constituent with original premise. We first report augmentation results with constituent with original premise. We see a modest, nonsignificant increase in HANS performance, as seen in Figure 6 (baseline 0.57, large 0.58). What is more prominent than an increase in HANS accuracy is the reduction of variance. Although it was

not discussed in detail in their paper, augmentation from Min et al. (2020) decreases variance in HANS performance documented by McCoy, Min, and Linzen (2019). The same trend is also observed here, most apparently shown in Figure 6.

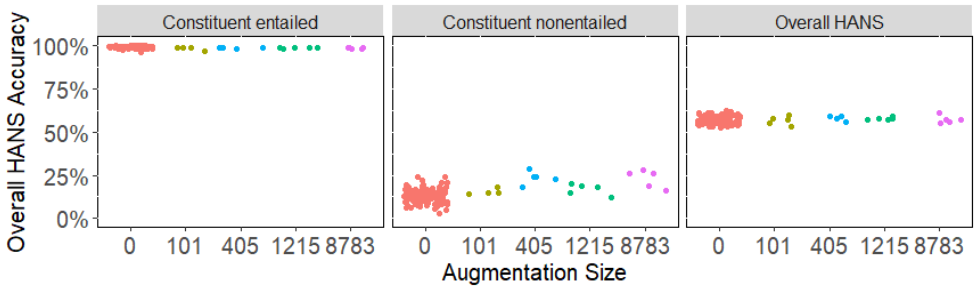


Figure 6
HANS performances on BERT fine-tuned on MNLI augmented with different sizes of the constituent with original premise augmentation dataset.

We next look into a subset of HANS examples that are inconsistent with the constituent heuristic that this augmentation set targets. Here, even a small augmentation of roughly 100 examples is enough to bump performance from the baseline levels (baseline 0.13, small 0.16). All four sizes report improvement over baseline levels (medium 0.24, large 0.17, extra-large 0.23) as seen on Figure 7. But interestingly, medium augmentation (n = 405) reports the best improvement. This pattern is also seen in Section 3.2, where they found that medium augmentation using inversion with transformed hypothesis was the most effective over many large-size (n = 1215) augmentations. An explanation is not immediately clear. We suspect it may be an artifact of randomly sampling directly from the generated dataset to create augmentation datasets, rather than iteratively subsetting which ensures that each augmentation is a subset of a larger dataset. It is possible that under this methodology, the medium augmentation sets contain more informative examples than do the large augmentation sets. However, this explanation is likely insufficient to explain the prevalence of this behavior across many augmentation strategies and often-poor performance of models trained with extra-large augmentations. We identify possible causes for such behavior. Our method of sampling augmentation data was not hierarchical–the small augmentation dataset was not a subset of the medium augmentation dataset, and so on. This way, a dataset with less examples could contain more helpful or informative examples. Alternatively, decreasing marginal return on additional augmentation data, most significantly demonstrated in the inversion with the transformed hypothesis augmentation from Figure 5 may eventually hurt performance, possibly due to overfitting. This calls for further study on informative and uninformative (and even disinformative) examples and overfitting to augmentation examples.

Within the HANS examples that are inconsistent with the constituent heuristic, there is the embedded under verb construction. An example with this construction is as follows:

(9) a. **Premise:** The tourists said that the lawyer saw the banker.
b. **Hypothesis:** The lawyer saw the banker.
c. **Label:** Non-entailment

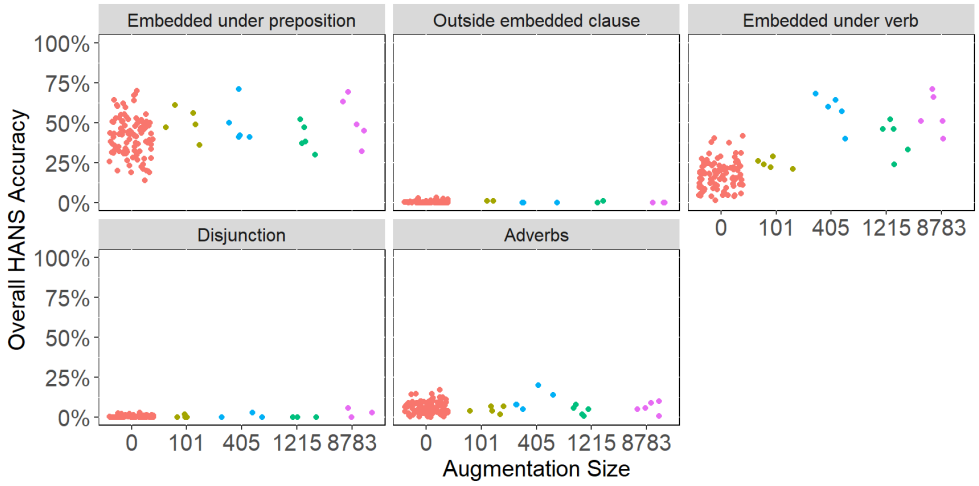


Figure 7
Performance on HANS examples that are inconsistent with the constituent heuristic for BERT fine-tuned on MNLI with different sizes of the constituent with original premise augmentation dataset.

Hypotheses for our augmentation data were generated by searching for sentences in MNLI with clauses embedded under a verb, and most closely resembles hypotheses from this HANS construction. Thus, we anticipate significant improvement in performance on this construction. Indeed, we see a prominent jump in accuracy from baseline (0.17) to all four sizes of the dataset (small: 0.24, medium: 0.58, large: 0.40, extra-large: 0.56). Despite the abstract similarity in augmentation data and the embedded under verb construction, such improvements are not insignificant. Echoing the argument from 3.2, the strategy is similar on an abstract level to these HANS examples, but the two should be considered different distributions, due to differences in vocabulary and syntactic properties. BERT was able to generalize syntax relevant to this construction from the augmentation dataset.

Another construction we see significant improvement is the embedded under preposition construction, with non-entailing sentence pairs like ("Unless the senators ran, the professors recommended the doctor." $\nrightarrow$ "The senators ran.").

Generalization to examples targeting different heuristics in HANS was minimal.

3.4.2 Constituent with transformed hypothesis. We now look at the effects of augmentation using constituent with a transformed hypothesis. First, we note that the number of examples in the extra-large augmentation dataset with transformed hypotheses is slightly over 3000, compared to almost 9000 with original premise. In Section 3.2, we found that augmentation examples with transformed hypotheses were more effective than augmentation examples with original premises. That is also the case with this augmentation strategy. As seen in Figure 8, jump in HANS performance is more apparent, with extra-large augmentation posting 0.60 accuracy (baseline 0.57; constituent with original premise, extra large 0.57).

Effects of augmentations are more visible within HANS example that are inconsistent to the constituent heuristic, shown in Figure 9. Even a small augmentation (n = 101) is able to improve accuracy from 0.13 (baseline) to 0.26. Here, the large dataset (n =

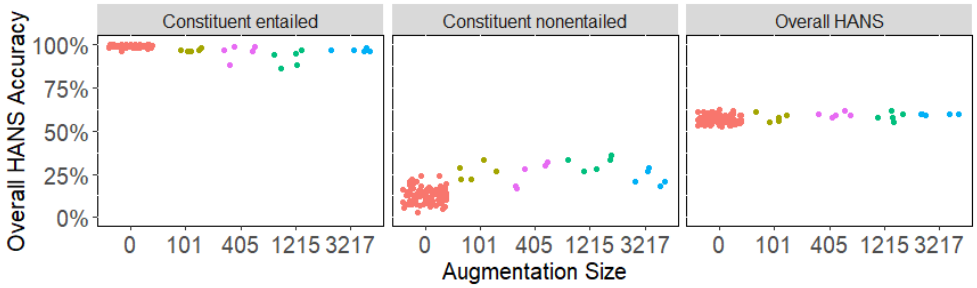


Figure 8
HANS performances for BERT fine-tuned on MNLI with different sizes of the constituent with transformed hypothesis augmentation dataset.

1215) generated the most significant increase in accuracy with 0.31. Here we see another example of a larger dataset inducing worse generalizations than a smaller dataset.

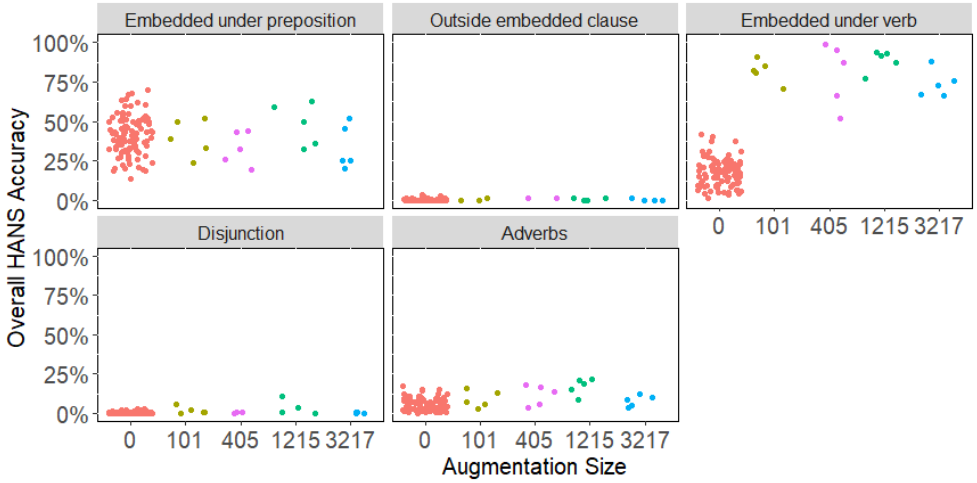


Figure 9
Performance on HANS examples that are inconsistent with the constituent heuristic for BERT fine-tuned on MNLI with different sizes of the constituent with transformed hypothesis augmentation dataset.

Finally, we look at the embedded under verb construction within HANS. The premises of examples with a transformed hypothesis too closely resemble those from the HANS examples, in addition to the hypotheses. Thus, we anticipate a greater level of improvement here than what we saw before, with augmentation using constituents with an original premise. Such anticipation proves to be true. The small augmentation (n = 101) is sufficient to boost BERT's performance on examples with this construction to 0.82, up from baseline's 0.17. The large augmentation of this kind is the most effective, with accuracy at 0.89.

Syntactic information learned from these counterexamples generalize to other constructions and heuristics. Within the same heuristic, we see notable improvements in the sentences with adverbs construction, and the embedded under preposition constructions. Within the subsequence heuristic, we see improvements in the PP on subject

and relative clause on subject constructions. Finally, generalization to the lexical overlap heuristic is considerable as well, reporting a 0.38 accuracy after the extra-large augmentation, up from a 0.28 baseline accuracy. Unsurprisingly, we see improvements in all four constructions within the lexical overlap heuristic, with the exception of passive sentences.

3.5 Discussion

In this section, we discuss solutions to MNLI and BoolQ models that have adopted heuristics. Because the heuristics are known, syntactic augmentation can be an effective method to counter them. We created counterexamples to the lexical overlap heuristics by inverting the subject and the object of a transitive sentence, and augmented the MNLI training set with the augmentation dataset. We found that not only did augmentation increase HANS performance in the subject-object swap construction which most closely resembles the augmentation data, but induced syntactic sensitivity that generalized to sentence constructions within lexical overlap and even to subsets of HANS targeting other heuristics. Moreover, syntactic augmentation is effective in reducing instability in out-of-distribution generalization reported in Section **??**.

In addition to augmentation targeting the lexical overlap heuristic, we design one targeting the subsequence heuristic, and create another augmentation dataset targeting the constituent heuristic. The augmentation is effective and increases HANS performance in the embedded under verb construction that most closely resembles the augmentation data. It generalizes to other constructions targeting the constituent heuristic, but fails to generalize to other heuristics like the inversion augmentation dataset. This may be due to the hierarchical nature of the heuristics, where the constituent heuristic is a subset of the subsequence heuristic, and the subsequence heuristic subset of the lexical overlap heuristic. Thus, syntactic sensitivity to word order improves HANS performance across all heuristics, but counterexamples to the constituent heuristic are only effective within the heuristic.

Because the heuristics from MNLI transferred to BoolQ models and improved their test set accuracy, we anticipated that syntactic knowledge from augmentation to MNLI will also transfer. This hypothesis predicts that training an augmented MNLI model on BoolQ will perform much better than a baseline BoolQ model in QA-HANS. However, this was not the case, as augmentation transfer models (0.53) performed similarly on HANS than BoolQ baseline models (0.53). The discrepancy may be due to the small number of augmentation examples, as effects of transfer on BoolQ test set accuracy was meager in small-MNLI and MNLI-last models. Given a much larger additional pretraining on augmentation-like examples, such syntax may transfer to BoolQ. Another yet unlikely reason due to poor transfer may be the structural difference between BoolQ and MNLI datasets. While an MNLI examples can easily be transformed into BoolQ and vice versa as discussed in Section 1.1, BoolQ examples contain its elements in the order of (question, passage, answer), while MNLI examples in the order of (premise, hypothesis, label). Because the transformation maps the question to the hypothesis and passage to the premise, this may cause some confusion in statistical learners such as BERT.

We augment our fine-tuning datset with adversarial examples to the target heuristics via simple transformations of training dataset examples. Other forms of providing additional information are effective as well. Additional fine-tuning with forgettable examples improve out-of-distribution generalization measured by HANS by almost 20%p (Yaghoobzadeh et al. 2019) over unaugmented baseline (McCoy, Min, and Linzen 2019),

and providing additional semantic (Cengiz and Yuret 2020) and syntactic (Moosavi et al. 2020) role labeling is also effective. A surgical, iterative approach to heuristics by providing one set of adversarial examples to remove one bias at a time has also been proposed (Nie et al. 2020). Generating synthetic augmentation data for augmentation is also effective outside of natural language classification tasks, in natural language generation (Shah, Schuster, and Barzilay 2020; Fadaee, Bisazza, and Monz 2017) and computer vision (Shorten and Khoshgoftaar 2019; Fawzi et al. 2016).

Another popular method is ensemble training, in which a deliberately biased model that was made especially vulnerable to adopting heuristics is built prior to building a main, robust model (Clark, Yatskar, and Zettlemoyer 2019; He, Zha, and Wang 2019; Karimi Mahabadi, Belinkov, and Henderson 2020; Pang, Lin, and Smith 2019). The robust model is trained in ensemble with the biased model, so that the robust model is disincentivized to learn the heuristics and biases the biased model already has. Then, only the robust model is used for evaluation. This method is more bias-agnostic and general than augmentation–there is no need for human intervention to identify biases and create appropriate augmentation data, and generalization may happen beyond the domain of human awareness of biases in syntax or semantics. However, the ensemble method can create forgettables from the test set as models depart from heuristics, and can hurt test set performance (Clark, Yatskar, and Zettlemoyer 2019; Karimi Mahabadi, Belinkov, and Henderson 2020). A related confidence regularization method (Utama, Moosavi, and Gurevych 2020a) prevents the model from exploiting the heuristics by discouraging it from predicting the highest-probability label, which improves both in-distribution and out-of-distribution performance.

There are attempts to modify or add to the model structure to improve robustness to heuristics (Moradshahi et al. 2019; Tu et al. 2020; Kang and Hashimoto 2020; Zhou and Bansal 2020). A multitask learning approach that uses one model and different classifiers for each task is more robust than a model fine-tuned for just one task (Tu et al. 2020). Additional layers or models, like a pre-trained parser (Zhou and Bansal 2020) or a tensor product representation layer (Moradshahi et al. 2019) on top of BERT enhance the models' abilities to generalize to out-of-distribution examples.

Finally, simple methods like re-initializing in search of better seeds and longer fine-tuning are also effective ways to improve out-of-distribution generalization (Zhang et al. 2020).

4. Unpacking seed versus order from previous experiments

4.1 Background and previous experiments

In experiments described in sections 2 and 3, we run several restarts of a model to more accurately evaluate its architecture or augmentation strategy. Each restart was started from the same pre-training checkpoint but had different fine-tuning data shuffling and initialization. Similarly, Devlin et al. (2019) ran multiple several restarts to select the best model on development set because fine-tuning BERT-large was "sometimes unstable." Zhang et al. (2020) echo that model initialization is a major factor that affects performance and speed of convergence at various layers.

In our previous study (McCoy, Min, and Linzen 2019), we study this behavior and formalize BERT's empirically known instability: between the random restarts with NLI, despite stable in-distribution performance in MNLI, out-of-distribution performance on HANS varied significantly. We show that despite the same architecture and overall input model initialization and treaining order affect degrees of heuristics reliance and

thus robustness: overall HANS performance varied between 0.53 and 0.63, but variance was much larger in non-entailed subsets of HANS. For instance, the accuracy on the construction "sentence with prepositional phrases," which are non-entailing despite lexical overlap, ranged between 0.04 and 0.77. However, such instability was not the case for performance on entailing sentences. Across all but two (13 out of 15 sentence constructions with entailment, minimum accuracy of all 100 instances were above 0.92.

In a follow-up study (Min et al. 2020), we observe that while syntactic augmentation reduces variance across model initialization and dataset order, it was still significant. There is variance across accuracy on both entailed and non-entailed sentence pairs. Upon preliminary analysis, we find that for the case of sentence pairs with lexical overlap, accuracy on entailed and non-entailed sentences are negatively correlated. We interpret a model's lower accuracy on entailed sentence pairs and higher accuracy on non-entailed as its robustness to heuristics that links lexical overlap and entailment. Thus, we argue that seed and order affect models' reliance on, as well as their ability to mitigate reliance on, heuristics. That is, seed and order affect performance in low-resource environment (non-entailing pairs in this case) and also models' ability to learn from new data and generalize.

Both seed and order are hyperparameters of interest as they affect generalization, robustness, etc. But, observations so far mix model initialization with dataset order in their experiments and analyses. In this section, we break down the effects of seed and order in MNLI models.

4.2 Experiments

To break down the effects of seed and order, we train three groups of MNLI models. Across the models in the first group, both model seed and MNLI order vary. Across the models in the second group, only model seed varies, while in the third group, only MNLI order varies.

We observe the models' in-distrubtion generalization to unseen examples in MNLI, and out-of-distribution generalization to HANS throughout training, for 3 epochs. Within HANS, we focus on examples that target the lexical overlap heuristic (i.e. sentence pairs that contain lexical overlap and are either entailing or non-entailing).

We fine-tune 27 iterations of BERT fine-tuned on MNLI, with 9 iterations with varying order of the training set, 9 iterations with varying classifier initialization, and 9 iterations with varying order and initialization. Each iteration of the model was evaluated on both MNLI development set (in-distribution), and subset of HANS targeting the lexical overlap heuristic (out-of-distribution), at training step 1, 10,000, and every subsequent 10,000 steps for a total of approximately 140k steps. An ideal model with perfect performance is illustrated in Figure 10.

We see some similarities across all three groups. First, we observe that in-distribution generalization, measured by MNLI accuracy, quickly converges and is consistent. By 10,000 training steps (320,000 examples with a batch size of 32), MNLI accuracy is consistently around 79%, within 4-5%p of MNLI accuracy of a fully trained model. After that, accuracy quickly plateaus and in-distribution generalization stabilizes.

Second, accuracy on examples consistent with the heuristic (entailing sentence pairs that agree with the heuristic) peaks early, then slowly decays. By 10,000 training steps, their reliance on the lexical overlap heuristic is conspicuous regardless of the direction of their initial bias after 1 training step. All 27 iterations score above 0.97 in entailment

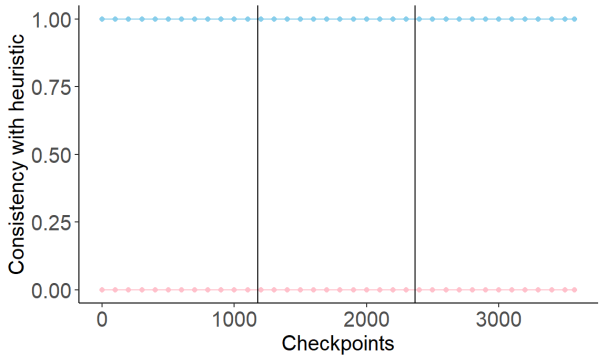


Figure 10
An ideal model with perfect entailment and non-entailment performances. Blue lines indicate accuracy one xamples consistent with the lexical overlap heuristic, and pink lines indicate 1-(accuracy on examples inconsistent with the lexical overlap heuristic).

examples, and below 0.03 in non-entailment examples, suggesting heavy reliance on the lexical overlap heuristic in the early stages of training.

Finally, accuracy on examples inconsistent with the heuristic (non-entailing sentence pairs that are counterexamples to the lexical overlap heuristic) are near-zero in early stages of training but increase over time, especially after the first epoch, after around 50,000 training steps. While accuracy on non-entailment examples increases, accuracy on entailment examples remains high despite a small decay. At the end of fine-tuning, non-entailed lexical overlap performance is not yet stabilized, strengthening the argument that a much longer fine-tuning is needed (Zhou et al. 2020) and out-of-generalization analyses under the current implementation of 3-epoch fine-tuning (McCoy, Pavlick, and Linzen 2019; McCoy, Min, and Linzen 2019) may be premature.

However, we were unable to identify any clear patterns that differentiate each of the three groups from the rest. All three groups, each with varying seed, varying order, and varying seed and order, exhibit similar behavior. This suggests that between seed and order, there there is no clear dominating factor that affects model behavior. Both hyperparameters play a role in determining a model instance's degree to which it relies on heuristics, and thus its out-of-distribution performance. We discuss implications of this behavior in the discussion section below.

4.3 Discussion

In this section, our experiments fail to identify a clear dominating factor between seed and order. We observe that both are important hyperparameters that affect properties like performance, degree of heuristic reliance(at the end of 3-epoch fine-tuning), and speed of convergence. It has been shown that model instances vary in robustness (measured by out-of-distribution performance on HANS) across different seeds with fixed order (Zhou et al. 2020; McCoy, Pavlick, and Linzen 2019; Devlin et al. 2019). But, they also vary in robustness across differently ordered training set with fixed seed. This suggests that dataset order is as important of a hyperparameter as seed–even though a collection of model instances trains on the same training *set*, the order in which its elements are arranged exercises a nontrivial effect.

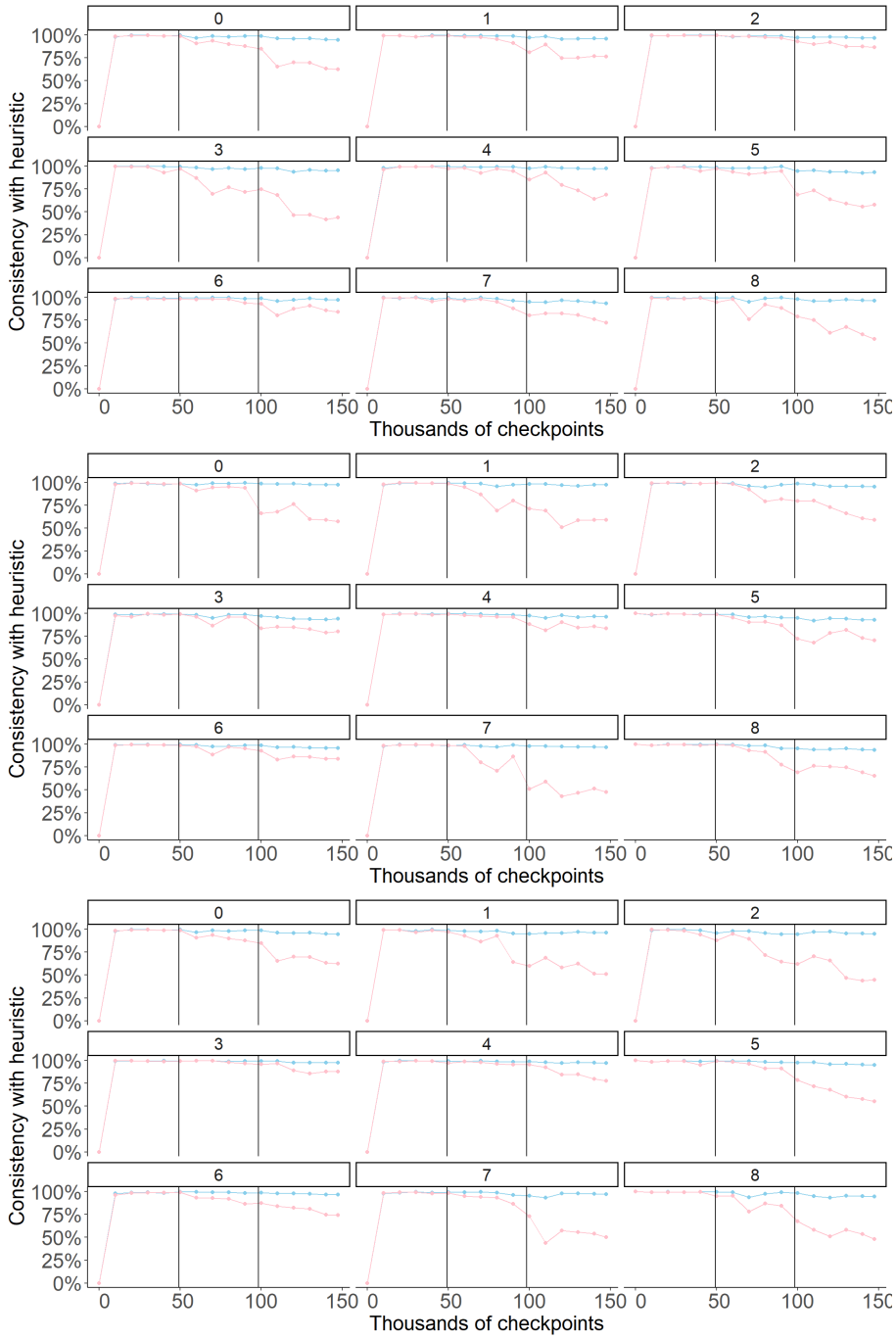


Figure 11
HANS lexical overlap performances of BERT finetuned on BoolQ with varying order only, varying seed only, and varying seed and order (top to bottom). Blue lines indicate accuracy on examples consistent with the lexical overlap heuristic, and pink lines indicate 1-(accuracy on examples inconsistent with the lexical overlap heuristic). Each black horizontal line indicates the end of an epoch. Models were fine-tuned for three epochs.

This may be attributable to batching. When we vary the dataset order, although the examples that comprise the dataset remain the same, the batches they form differ. In the second group of model instances, in which only seed varies, we see clearly that some batches of examples are more influential than others. We focus on change in accuracy in counterexamples to the lexical overlap heuristic between the 90,000th and 100,000th training steps. In all 9 instances within the group, we see positive gains, and in 5 out of 9, the performance gain in the interval is the greatest throughout the 3 epochs of fine-tuning. However, in groups 1 and 3, a similar jump either occurs somewhere else (where a similarly influential batch is formed), or does not occur at all (no similarly influential batch is formed). Different dataset orderings generate different varieties of sets of batches, which then lead to significant differences in model behavior.

In addition, this behavior also suggests there could be no "magic seed" that encodes good linguistic information, as such seed would have been more resistant to effects from variations across order. On the other hand, this also suggests future research could benefit from finding and training models with dataset order that is more conducive to robustness, just as done with seed (Zhou et al. 2020).

The experiments from this section also offer insight outside of seed and order. We observe that robustness to overlap heuristics improves throughout epochs, without plateauing within the 3 epoch recommended by Devlin et al. (2019). this echoes findings by Zhang et al. (2020) where longer fine-tuning improved BERT's performance across various tasks, and provides a bridge between the Representational Inadequacy and Missed Connection hypotheses, where training parameters were suboptimal–with fine-tuning only lasting 3 epochs.

Thus, we echo other works in their call for longer fine-tuning until convergence for a more accurate evaluation of model architectures and datasets. Not only does longer fine-tuning improve performance, but stopping fine-tuning before a model reaches convergence may undeservingly underscore its instability, given a potentially stochastic nature of its path toward convergence. If fine-tuning until convergence is impossible or even unrealistic, multi-seed evaluation throughout training will be necessary to accurately gauge a model's behaviors.

5. How specific is this to MNLI?

5.1 Background

Earlier in this paper, we observe that BERT-based MNLI models adopt heuristics and thus report poor out-of-distribution generalization. Consequently, we found that syntactic data augmentation is an effective method to mitigate such behavior.

In addition, we attempted to further break down the effects of varying seed and training dataset order between model initializations and concluded that there is no clear dominating factor. That is, both the dataset (MNLI) and the architecture (BERT) contribute to the models' behaviors such as inclination toward heuristics, Furthermore, we reinforce findings by Zhang et al. (2020) and Mosbach, Andriushchenko, and Klakow (2021) that longer fine-tuning improves performance b showing that out-of-distribution generalization significantly improves each epoch, and does not pleateau within the 3 epochs, a common length of MNLI fine-tuning (Devlin et al. 2019; McCoy, Pavlick, and Linzen 2019; McCoy, Min, and Linzen 2019; Utama, Moosavi, and Gurevych 2020b; Min et al. 2020).

Now, what about other tasks? Are our observations–heuristic adoption, poor out-of-distribution generalization, effects of seed and order, and benefits of longer-fine-tuning–

NLI
Premise: The managers near the scientist shouted.
Hypothesis: The scientist shouted.
Label: Non-entailment
QA
Passage: The managers near the scientist shouted.
Question: Did the scientist shout?
Answer: No

Table 5
An NLI example can be transformed into a QA example by transforming the hypothesis into a question, and mapping the entailment and non-entailment labels to the yes and no answers.

specific to MNLI? Or do BERT-based models show similar behaviors across different tasks and datasets? We observe whether similar phenomena persist in another task–Boolean QA, a question answering task similar to MNLI.

5.2 Datasets

In this section, we discuss two datasets: BoolQ, a Boolean QA dataset, and QA-HANS, a HANS-like dataset reformatted as a QA dataset.

BoolQ Clark et al. (2019) is a Boolean QA dataset that consists of a passage, question, and a yes/no answer. The dataset is much shorter than MNLI at almost 9,000 sentence pairs, and is obtained from full-text Google queries and Wikipedia excerpts, if Wikipedia was one of top 5 search results for the query.

BoolQ is structurally similar to NLI, so NLI examples can be easily transformed into BoolQ examples, and vice versa. NLI's premise can act as BoolQ's passage, while NLI's hypothesis can be transformed into a Boolean question to serve as BoolQ's question. Then, the output labels "entailment" and "non-entailment" can be mapped to BoolQ answers "yes" and "no," respectively, as seen in Table 5.

However, while MNLI and its connection between lexical overlap and the entailment label as seen in Section 2.3, BoolQ presents a different distribution of labels against lexical overlap.

We analyze the BoolQ dataset in the same manner as in Section 2.3. A histogram of BoolQ answers by overlap rate shows that in BoolQ training set, there is not a clear correlation between overlap rate between rate of lexical overlap and ratio of examples with yes answers–across all overlap rates, there are more “yes” than “no” answers. This may encourage statistical learners to be biased toward the "yes" answer, and reward similarly biased models. We also observe that the dataset includes more high-overlap rate examples than low-overlap rate examples, further unbalancing the training and evaluation dataset.

Although MNLI and BoolQ datasets are structured similarly there is one major difference. MNLI examples have three labels–entailment, neutral, and contradiction–the latter two of which are masked into non-entailment in our following experiments. Models trained on MNLI output one of the three labels, which is then mapped to the masked labels. On the other hand, BoolQ has two answers that are directly output by the models.

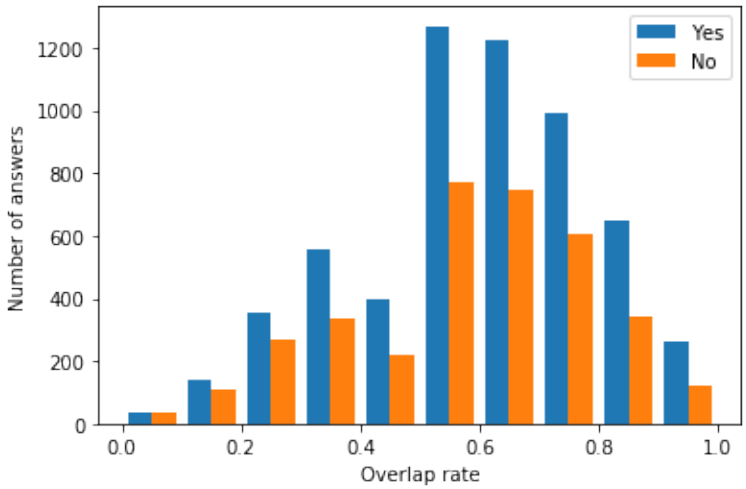


Figure 12
Histogram of BoolQ answers by overlap rate.

QA-HANS is a HANS-like (McCoy, Pavlick, and Linzen 2019) evaluation set we propose for Boolean QA's out-of-distribution evaluation. It is obtained by using the NLI-to-BoolQ transformation on HANS, described above. The dataset holds same specifications as HANS, with 5,000 examples for each heuristic and answer label. It comprises of 2 answer labels (yes and no) and 3 heuristics (lexical overlap, subsequence, constituent), for a total of 6 categories and 30,000 examples. An example of NLI-to-BoolQ transformation is illustrated in (11-e).

HANS examples (1) and (2) from (2-c) are transformed to the following to form QA-HANS examples. The transformation includes modifying the hypothesis to a question, and changing the order of (premise, hypothesis, label) to (question, passage, answer).

(10) a. **Subcase:** Understood argument
b. **Template:** Did the N_2 V_{pres}? The N_1 P the N_2 V_{past}.
c. **Question:** Did the actor run?
d. **Passage:** The lawyer by the actor ran.
e. **Answer:** No.

(11) a. **Subcase:** PP on subject
b. **Template:** Did the N_1 V_{pres}? The N_1 V_{past} the N_2.
c. **Question:** Did the author read?
d. **Passage:** The author read the book.
e. **Answer:** Yes.

5.3 Experiments

To find out whether our previous observations are specific to MNLI, we ask the same questions we asked in the previous sections. First, we reproduce experiments from Clark et al. (2019) as sanity checks to ensure our setup, and establish out-of-distribution baselines with the QA-HANS dataset.

We follow Clark et al. (2019) and fine-tune BERT on the BoolQ training set for 5 epochs to create the baseline BoolQ model. We successfully replicate their in-distribution performance, with the BoolQ test set performance averaging at 0.73 over 40 runs. The test set performance was consistently at a similar level across all runs as standard deviation was only 0.01. This behavior where models consistently achieve high test set accuracy is similar to that of BERT models trained and tested on MNLI.

BERT trained on MNLI and BERT trained on BoolQ exhibit another resemblance– poor and unstable out-of-distribution generalization. The out-of-distribution performance baseline measured with evaluating each of the 40 runs on QA-HANS. While not demonstrating the lexical overlap, subsequence, and constituent heuristics, still display poor generalization to new datasets, with overall QA-HANS accuracy standing at 0.53, slightly above chance. Out-of-distribution generalization was also unstable, with standard deviation ranging between 0.32 and 0.38 for each category of QA-HANS. There is no apparent bias toward either labels. Accuracy for passage-question pairs whose gold answer is "yes" was 0.53, while accuracy for those whose gold answer is "no" was 0.52. Average accuracy and standard deviation across the 40 BoolQ baseline runs test set and QA-HANS is shown in Figure 6, and we are able to characterize the BoolQ baseline models as lacking particular bias yet highly variant.

In addition to a set of baseline models, Clark et al. (2019) show that using MNLI as additional pre-training improves recurrent and transformer-based BoolQ models. In particular, recurrent models improve on the BoolQ test set by 6%p, from 0.69 to 0.75, with transfer from MNLI. We replicate the behavior in BERT by providing additional pre-training in MNLI on a BoolQ model for 3 epochs. We implement this by fine-tuning a BERT-based MNLI model on BoolQ. We trained 10 instances of the BoolQ models with MNLI transfer, each with different seed and order of the BoolQ training set. Their average BoolQ test set accuracy was 0.79, up $6\%p$ from 0.73 of baseline BERT models. We are able to verify that additional pre-training with MNLI improves in-distirbution generalization.

Will the same hold for out-of-distribution generalization? We also evaluate effects of MNLI transfer on QA-HANS. Unlike the improvement in in-distribution generalization, MNLI transfer models perform worse on QA-HANS than BoolQ baseline models slightly, but with statistical significance (0.51, baseline 0.53; $p < 0.01$). Despite the small difference, MNLI transfer models demonstrated dramatically different behavior in their QA-HANS performances. Two notable characteristics seen in baseline BoolQ models' QA-HANS performances disappeared: identical consistency to heuristics and high variance. The MNLI transfer models showed bias toward the entailment label or the yes answer, with higher accuracies in passage-question pairs with yes answers and lower accuracies in pairs with no answers. Such bias is likely picked up from the transfer. This conspicuous change is intuitive and likely due to the relative sizes of MNLI and BoolQ training sets, at $297k$ and just above $9k$. Drastic reduction of out-of-distribution generalization variance across the board the is less intuitive due to BERT-based MNLI models' high instability in lexical nonentailed performance but is nonetheless a result of the additional pre-training. Other characteristics of the BERT-based MNLI models, such as near-zero subsequence nonentailed performance, did not transfer.

Overall, the additional pre-training with MNLI yields greater in-distirbution generalization in the BoolQ dataset, without increasing robustness in the boolean QA task. How does this happen? We later offer an explanation of this phenomenon in the Section 5.4.

So far, we have replicated experiments of Clark et al. (2019) while measuring out-of-distribution generalization for the baseline BoolQ and MNLI transfer models. We also

Label	Target heuristic	Baseline Average	Baseline Stdev	MNLI transfer Average	MNLI transfer Stdev	Aug transfer Average	Aug transfer Stdev
BoolQ test set		0.73	0.01	0.79	0.00	0.79	0.00
Yes	L	0.53	0.36	0.73	0.04	0.71	0.05
	S	0.61	0.35	0.71	0.05	0.74	0.07
	C	0.47	0.33	0.71	0.03	0.80	0.03
No	L	0.45	0.38	0.31	0.03	0.34	0.03
	S	0.48	0.34	0.46	0.05	0.42	0.03
	C	0.63	0.32	0.11	0.04	0.15	0.03
Overall QA-HANS		0.53	0.02	0.51	0.01	0.53	0.01

Table 6
BoolQ and QA-HANS performances of with and without transfer. Number of model instances for each type was 40 for baseline, 10 for all others. *L, S, C* refer to the lexical overlap, subsequent, and constituent categories of QA-HANS.

make a novel observation that while additional pre-training with MNLI on a BoolQ baseline model improves test set performance, the improvement does not generalize to out-of-distribution challenge sets, and instead learns biases that are detrimental to generalization. The biases include the lexical overlap heuristic, the subsequence heuristic, and the constituent heuristic, described in Section 2. A natural idea that follows is to attempt to mitigate the biases learned from the additional pre-training by syntactically augmenting the MNLI dataset as done in Section 3.2.

To see whether syntactic data augmentation improves out-of-distribution generalization of MNLI transfer models, we substitute the additional pre-training with MNLI with additional pre-training with augmented MNLI, while keeping all else the same. That is, we take a BERT model trained on augmented MNLI for 3 epochs, then train the model on BoolQ for 5 epochs. We expect the augmentation to have a similar effect to that we saw in Section 3.2–little change in in-distribution generalization performance, and significant improvement in out-of-distribution generalization, especially in counterexamples to the lexical overlap heuristic.

The results for models with this setup, which we dub *Aug transfer*, are also shown in Table 6. Similarly to how syntactic augmentation affects MNLI models, there was little change in in-distribution generalization (*MNLI transfer*: 0.79, *Aug transfer*: 0.79. On the other hand, contrary to our expectations, syntactic augmentation to BoolQ models did not improve out-of-distribution generalization. Although overall QA-HANS accuracy improves by $2\%p$, a simple look into accuracies for each category shown in Table 6 reports no significant improvement on examples with no answers (counterexamples to heuristics). We observe that unlike in NLI, syntactic augmentation in Boolean QA has little effect in mitigating the lexical overlap heuristic.

Next, we study the effects of seed and order in Boolean QA. Similarly to the experiments in Section 4, where we found that there was no prevalent factor between the two, we train three groups of BoolQ models. Each group contains six models. Across the models in the first group, both model seed and BoolQ order vary. Across the models in the second group, only model seed varies, while in the third group, only BoolQ order varies. Performance is recorded every 100 training steps, for a total of 35 intermediate

checkpoints during fine-tuning, which corresponds to around 7 per epoch. The results are shown in 13, which we discuss further in the discussion section.

We hope again to disentangle effects of seed from effects of dataset order, but we again witness no clear effects each exerts. However, we observe two other significant behaviors.

First, we see here more explicitly the relationship between accuracies on entailed and non-entailed examples. Although not exactly, accuracy on entailed examples is almost identical to the complement of accuracy on non-entailed examples ($ent_{acc} = 1 - non_{acc}$), reinforcing our finding in Section 5.3. Not only for every model but also at every single checkpoint, identical consistency with heuristics is present.

Second, we witness the existence of super-influential examples within the training set. Even with only a quick look at the six runs with varying seeds, we observe conspicuous "dips" present at the same training step in all six models. In all six iterations with varying initializations, there are a few neighborhoods of training steps that are much more influential than others. This suggests that such neighborhoods contain many influential examples, and in turn suggest that there are high-influence examples that affect the model much more than others. This hypothesis holds with what we see from the six runs with varying orders, where the "dips" are scattered throughout training.

Finally in this section, we study the effects of longer fine-tuning in BERT-based BoolQ models. Devlin et al. (2019) suggest BERT be fine-tuned for 3 epochs for downstream tasks and Clark et al. (2019) fine-tune BERT on BoolQ for 5 epochs. We observe 40 instances of BoolQ models with varying seed and training set order, fine-tuned for 20 epochs, whose performance is recorded every 500 training steps. We expected the BoolQ models to improve in QA-HANS performance as training progressed, echoing MNLI models' behavior we observed in 4, where longer fine-tuning resulted in increased robustness. We discuss the results, shown in 14 in the discussion section.

5.4 Discussion

In this section, we expand beyond MNLI in studying BERT's behaviors to BoolQ, a Boolean QA dataset. BoolQ has similarities to MNLI but is framed in a question-answer format. We study whether behaviors we observed in our BERT-MNLI experiments in previous sections persist with BoolQ. Does syntactic augmentation increase robustness in QA heuristics? Do varying seed and order both affect BoolQ models like they do MNLI models? And will longer fine-tuning improve robustness? During our discussion, we also provide an explanation on hot additional pre-training on MNLI improves in-distribution performance despite lack of out-of-distribution (QA-HANS) performance.

We first set up BoolQ baselines by replicating results from Clark et al. (2019) by training a BERT-based BoolQ model, as well as a model with additional pre-training with MNLI. We also establish a QA-HANS baseline, in which we observe no significance reliance on any of the three syntactic heuristics.

Clark et al. (2019) along with our replication establish that additional MNLI pre-training improves in-distribution generalization measured by BoolQ test set. We investigate how the additional pre-training affects the BoolQ model. To understand what is happening better, we also measure the additionally pre-trained model's QA-HANS performance. Interestingly, we see no improvement in out-of-distribution generalization, despite clear in-distribution generalization improvement. That is to say the additional pre-training does not contribute to more effective understanding of passages or questions in the BoolQ dataset, but rather helps the model exploits a property of the dataset that offers little meaningful connection to the task of Boolean question answering.

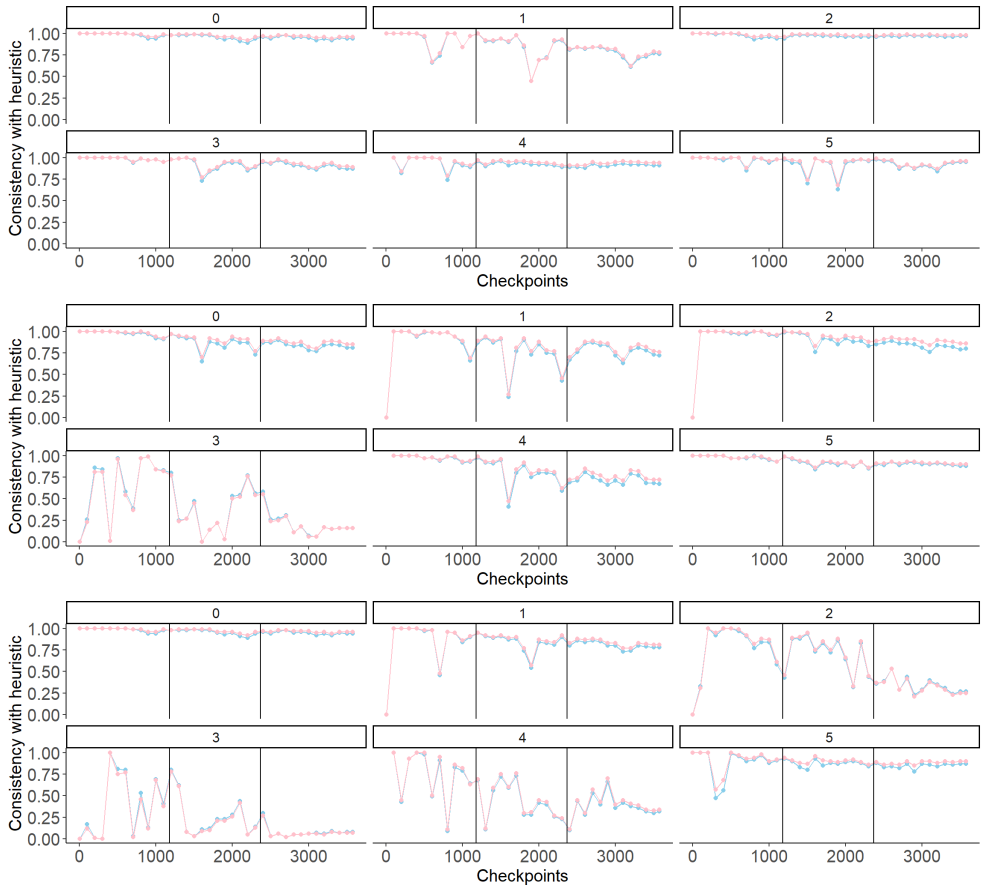


Figure 13
QA-HANS lexical overlap performances of BERT fine-tuned on BoolQ with varying order only (Figure 8), varying seed only (Figure 9), and varying seed and order (Figure 10). Blue lines indicate accuracy on examples consistent with the lexical overlap heuristic, and pink lines indicate 1-(accuracy on examples inconsistent with the lexical overlap heuristic). Each black horizontal line indicates the end of an epoch. Models were fine-tuned for three epochs.

How does this happen? An analysis of the MNLI dataset, by McCoy, Pavlick, and Linzen (2019) and in Section 2.3 shows that within the dataset, there is correlation between lexical overlap and the entailment label. Such property leads statistical learners like BERT to adopt the lexical overlap heuristic (McCoy, Pavlick, and Linzen 2019; McCoy, Min, and Linzen 2019). A similar phenomenon happens to the BoolQ model with additional pre-training with the MNLI dataset. In Table 6, we see that the additional pre-training with MNLI transfers its inclination for the BoolQ analog of the entailment label–the yes answer. With a gentle reminder of a BoolQ distribution analysis from Section 5.2 that the BoolQ dataset includes many more yes answers than no answers, it is no surprise that such bias would improve the model's performance on the BoolQ test set by shifting the model's output to the yes answers.

We observed that while vanilla BoolQ models exhibit no reliance on the syntactic heuristics, MNLI transfers the lexical overlap heuristics to BoolQ models with additioanl pre-training with the MNLI dataset. We ask whether syntactic augmentation to

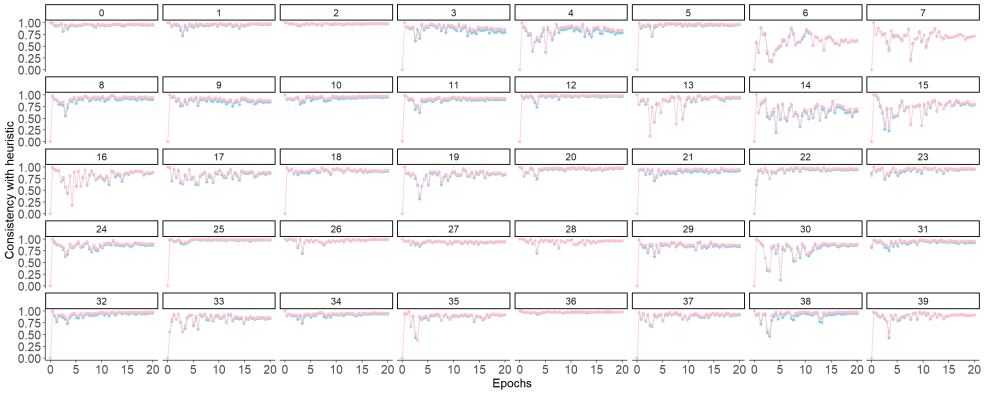


Figure 14
QA-HANS lexical overlap performances of 40 instances of BERT finetuned on BoolQ, across different initializations. Models were finetuned for twenty epochs. Blue lines indicate accuracy on examples consistent with the lexical overlap heuristic, and pink lines indicate 1-(accuracy on examples inconsistent with the lexical overlap heuristic).

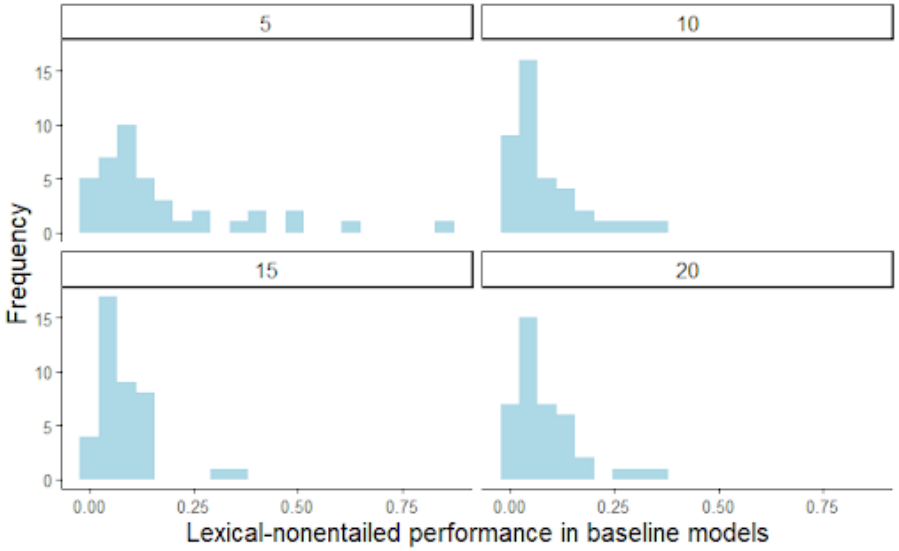


Figure 15
Distribution of QA-HANS lexical nonentailed performances in 40 instances of BERT finetuned on BoolQ, after 5.1, 10.2, 15.3, and 20 epochs

MNLI would mediate the heuristic like it did to MNLI models. We found that contrary to the augmentation effects to MNLI models, the augmentation was ineffective in mediating bias, and thus resulted in no improvement in out-of-distribution performance.

Next, we aim to disentangle the effects of varying seed and training dataset order in BoolQ models. We observe different behaviors from MNLI models, where we found no single dominating factor. With BoolQ models, seed seems to be more influential than order. There are initializations whose loss surface is rugged (with many local minima and maxima) and thus results in volatile performance changes throughout training, as

seen in runs 2 and 3, where within a single epoch, QA-HANS performance may vary by more than 0.5. While varying training set order can lead to more and less influential batches of training data, seed can determine whether the training process will be smooth (runs 0, 1, 4, 5) or volatile (runs 2, 3).

On the other hand, there are some common traits with the MNLI models as well. We find that some batches of examples are more influential than others. Similarly to what we found in MNLI, we were able to pick out one most influential set of batches that led to the steepest change in performance. In 5 of the 6 runs of fine-tuning BERT on BoolQ, the deepest valley in the performance graph is found at the same checkpoint, in the middle of epoch 2. The portion of the training dataset that corresponds to that checkpoint can be considered more influential than others.

Finally, with the effects of longer fine-tuning in MNLI models in mind, where it helped improve robustness measured by HANS performance. In BoolQ models, longer fine-tuning does not seem to improve robustness, unlike what is seen in MNLI. Consistency with heuristics over non-entailed and entailed QA-HANS examples is almost identical for every model. This behavior persisted for 20 epochs and showed no signs of change. It is unlike what we saw with MNLI, where longer fine-tuning improved robustness. Such behavior of BoolQ means that overall QA-HANS performance is capped at around 0.5, and longer fine-tuning will not improve out-of-distribution generalizability measured by the overall QA-HANS performance.

Our results indicate that findings in BERT-based MNLI models cannot be generalized to BERT-based BoolQ models, despite the datasets' similarities. This shows that discoveries in natural language understanding made with few experimental setups should carefully be examined before being used to derive a general conclusion. In addition, this behavior could indicate that despite the progress it has made on leaderboards and in empirical case studies, BERT is still dependent on surface features of the dataset such as lexical, n-gram, and label distributions, and again emphasizes the importance of rigorous evaluation methodologies.

6. General discussion

We examine MNLI and BoolQ models in their various stages of the PAID paradigm, and find methodologies within the paradigm that fuel progress unrelated to the desired human-like generalization. Where applicable, we propose modifications or additions to the methodologies to better align the paradigm with human-like generalization abilities.

First, MNLI and BoolQ datasets contain signals that induce bias in models. We find evidence for this in all three stages of the PAID paradigm. The most straightforward illustration of bias realizes in MNLI models as heuristics. In Section 2.2, we observe that all 100 instances of BERT fine-tuned on MNLI adopt the lexical overlap, subsequence, and constituent heuristics. Without exception, all BERT-based models have managed to learn the heuristics, suggesting that they may be adopted during the fine-tuning stage. Bias is observable at the surface level in the datasets, as well. In Sections 2.3 and 5.2, we observe the imbalance in both MNLI and BoolQ datasets. In MNLI, complete lexical overlap (McCoy, Pavlick, and Linzen 2019) or high rates of it is correlated with the entailment label. It follows that low lexical overlap is associated with non-entailment labels–contradiction and neutral. While such imbalance may not directly trigger the lexical overlap heuristic, it still agrees that examples with high lexical overlap are likely to carry the entailment label.

During pre-training, the next sentence prediction objective focuses on sentence-pair representations, rather than representations at the lexical or the subsequence level. This may undermine the importance of word order and encourage heuristics too.

In Section 3.2, we augment MNLI to provide examples with high lexcial overlap with non-entailment labels. This syntactic augmentation helps balance out the MNLI dataset and serves as adversarial examples to the lexical overlap heuristic. From the experiments, we observe that with some exceptions (models struggle with passive sentences with or without augmentation–this may be attributed to insufficiency in pre-training) syntactic data augmentation mitigates the models' reliance on the lexical overlap heuristic and improves overall robustness to syntax in inference. Results from augmenting the MNLI dataset with counterexamples to the heuristics supports that while it may not be the sole cause of poor generalization, MNLI includes biased signals. Deep neural networks easily take advantage of such biased signals to make spurious connections and may choose to "learn the dataset" rather than the task, but known biased signals can be countered by adversarial, syntactic data augmentation.

Discuss heuristics in QA systems. BoolQ is also unbalanced. BoolQ models appear to adopt heuristics (but only in PyTorch implementations). Need additional research. Robustness to NLI transfers, but not augmentation. Anticipate QA-specific syntactic data augmentation will be effective.

BERT-based BoolQ models showed divergent results as only those implemented under the PyTorch framework adopt the lexical overlap heuristic. As the BoolQ dataset does not contain signals related to the heuristics, the heuristic is likely to have risen from BERT's pre-training. The BoolQ dataset is still unbalanced. It contains many more yes answers than no answers, and may encourage and reward models with bias toward the yes answers. This hypothesis is reinforced by results from MNLI transfer to BoolQ, where we find that with heursitics transfer, BoolQ models with additional pre-training on MNLI perform better on the BoolQ test set than baseline BoolQ models. Because we observed MNLI heuristics transfer to BoolQ, we anticipated that effects of MNLI augmentation would also transfer to BoolQ models. This was not true, as MNLI augmentation did not result in improvement in out-of-distribution generalization to QA-HANS. While we predict that QA-specific syntactic data augmentation will be effective, the effects will need to be confirmed by a future study.

A related problem within the PAID paradigm is the evaluation methodology that rewards models for making spurious connections and adopting heuristics. When models that adopt heuristics are evaluated on the test set, whose examples are drawn from the same distribution as those in the training dataset, they are undeservingly rewarded for adopting the heuristics that the training set suggest. This in-distribution evaluation naively gauges models on how well they learn the datasets, and is not a reliable method to indicate how well they perform on the task. Out-of-distribution evaluation is necessary to determine whether the model in question is able to generalize to new examples within the same task, like humans are able to. HANS (McCoy, Pavlick, and Linzen 2019) is able to measure robustness in NLI systems, while QA-HANS from Section 5.2 in QA systems.

Evaluation on generalization to out-of-distribution datasets presents another problem. We observe remarkable instability in out-of-distribution generalization for MNLI (Section 2.2) and BoolQ models (Section 5.3 across initialization and order in which fine-tuning data is sampled. While one may treat initialization as a hyperparameter that can be optimized (Zhang et al. 2020), we contend that such instability is a feature of the model and seeds that result in poor generalization abilities also need to be represented in model evaluation. We propose a twofold approach to this issue. First, models should

be analyzed on a large number of initializations (Sections 2.2 and 5.3). Due to instability, a small sample of initializations may be insufficient to represent the model architecture, training set, and other hyperparameters. Our analyses on 100 instances of MNLI models and 40 instances of BoolQ models, on the other hand, provide the big picture of how the models behave. Second, we advocate for longer fine-tuning (Section 5.3). We observe less reliance on the lexical overlap heuristic as training progresses in MNLI models, and less instability as trainign progresses in BoolQ models. This proposal is in agreement with other recent work (Zhou et al. 2020; Zhang et al. 2020).

7. Conclusion

In recent months, we have observed rapid growth in natural language understanding, fueled by scale in pre-trained models (Devlin et al. 2019; Raffel et al. 2020) and leaderboards (Wang et al. 2018, 2019). However, success measured by comparison between neural models' and human's performance on test sets is myopic, as neural models can adopt heuristics that excel in the test set, but fail on examples from a differently generated dataset (Agrawal, Batra, and Parikh 2016; McCoy, Pavlick, and Linzen 2019; Wang et al. 2017; Jia and Liang 2017). Our analysis on MNLI (Williams, Nangia, and Bowman 2018) and BoolQ datasets (Clark et al. 2019) show imbalance within datasets, which statistical learners like BERT (Devlin et al. 2019) may take advantage of. MNLI and BoolQ model behavior throughout fine-tuning show that while pre-training representation includes stimuli that incentivizes heuristics, poor and unstable generalization (McCoy, Min, and Linzen 2019) may be attributed to insufficient fine-tuning in currently popular implementations of BERT. Finally, we propose syntactic data augmentation as an effective method to mitigate known bias in natural language inference in NLI.

Our study contrasts behaviors of models subscribing to the PAID paradigm (Linzen 2020) to that of humans and proposes datasets (QA-HANS: Section 5.2, augmentation data: Sections 3.2, 3.3) and methodologies (out-of-distribution evaluation, longer fine-tuning, balancing of datasets, syntactic data augmentation) to close the gap between machine and man. While effective in assessing and improving generalization, reducing instability, and mitigating heuristics, the solutions we offer are by no means comprehensive and we predict that they are still far from what can generate human-like models. Thus, we call for architectures with superior generalization abilities and evaluation methodologies that can appropriately measure them to advance toward robust, human-like models.

Acknowledgments

We are grateful to Emily Pitler, Dipanjan Das, and the members of the Johns Hopkins Computation and Psycholinguistics lab group for helpful comments. Any errors are our own. This project is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. 1746891 and by a gift from Google, and it was conducted using computational resources from the Maryland Advanced Research Computing Center (MARCC). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation, Google, or MARCC.

References

Agrawal, Aishwarya, Dhruv Batra, and Devi Parikh. 2016. Analyzing the behavior of visual question answering models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language*

Processing, pages 1955–1960, Association for Computational Linguistics, Austin, Texas.

Bowman, Samuel R., 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 Empirical Methods in Natural Language Processing*, pages 632–642, Association for Computational Linguistics, Lisbon, Portugal.

Cengiz, Cemil and Deniz Yuret. 2020. Joint training with semantic role labeling for better generalization in natural language inference. In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 78–88, Association for Computational Linguistics, Online.

Clark, Christopher, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Association for Computational Linguistics, Minneapolis, Minnesota.

Clark, Christopher, Mark Yatskar, and Luke Zettlemoyer. 2019. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4069–4082, Association for Computational Linguistics, Hong Kong, China.

Cooper, Robin, Richard Crouch, Jan van Eijck, Chris Fox, Josef Genabith, Jan Jaspars, Hans Kamp, David Milward, Manfred Pinkal, Massimo Poesio, Steve Pulman, Ted Briscoe, Holger Maier, and Karsten Konrad. 1996. Using the framework.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Association for Computational Linguistics, Minneapolis, Minnesota.

Fadaee, Marzieh, Arianna Bisazza, and Christof Monz. 2017. Data augmentation for low-resource neural machine translation. *CoRR*, abs/1705.00440.

Fawzi, A., H. Samulowitz, D. Turaga, and P. Frossard. 2016. Adaptive data augmentation for image classification. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3688–3692.

Gururangan, Suchin, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, Association for Computational Linguistics, New Orleans, Louisiana.

He, He, Sheng Zha, and Haohan Wang. 2019. Unlearn dataset bias in natural language inference by fitting the residual. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 132–142, Association for Computational Linguistics, Hong Kong, China.

Jia, Robin 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, Association for Computational Linguistics, Copenhagen, Denmark.

Kang, Daniel and Tatsunori Hashimoto. 2020. Improved natural language generation via loss truncation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 718–731, Association for Computational Linguistics, Online.

Karimi Mahabadi, Rabeeh, Yonatan Belinkov, and James Henderson. 2020. End-to-end bias mitigation by modelling biases in corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8706–8716, Association for Computational Linguistics, Online.

Kim, Najoung, Roma Patel, Adam Poliak, Patrick Xia, Alex Wang, Tom McCoy, Ian Tenney, Alexis Ross, Tal Linzen, Benjamin Van Durme, Samuel R. Bowman, and Ellie Pavlick. 2019. Probing what different NLP tasks teach machines about function word comprehension. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 235–249, Association for

Computational Linguistics, Minneapolis, Minnesota.
Linzen, Tal. 2020. How can we accelerate progress towards human-like linguistic generalization? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5210–5217, Association for Computational Linguistics, Online.
McCoy, R. Thomas, Robert Frank, and Tal Linzen. 2018. Revisiting the poverty of the stimulus: hierarchical generalization without a hierarchical bias in recurrent neural networks.
McCoy, R. Thomas, Junghyun Min, and Tal Linzen. 2019. Berts of a feather do not generalize together: Large variability in generalization across models with similar test set performance.
McCoy, Richard T. and Tal Linzen. 2019. Non-entailed subsequences as a challenge for natural language inference. In *Proceedings of the Society for Computation in Linguistics*, volume 2, pages 358–360, Society for Computation in Linguistics, New York, New York.
McCoy, Tom, 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, Florence, Italy.
Min, Junghyun, R. Thomas McCoy, Dipanjan Das, Emily Pitler, and Tal Linzen. 2020. Syntactic data augmentation increases robustness to inference heuristics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2339–2352, Association for Computational Linguistics, Online.
Montague, Richard. 2008. *The Proper Treatment of Quntification in Ordinary English*, volume 49.
Moosavi, Nafise Sadat, Marcel de Boer, Prasetya Ajie Utama, and Iryna Gurevych. 2020. Improving robustness by augmenting training sentences with predicate-argument structures.
Moradshahi, Mehrad, Hamid Palangi, Monica S. Lam, Paul Smolensky, and Jianfeng Gao. 2019. Hubert untangles bert to improve transfer across nlp tasks.
Mosbach, Marius, Maksym Andriushchenko, and Dietrich Klakow. 2021. On the stability of fine-tuning {bert}: Misconceptions, explanations, and strong baselines. In *International Conference on Learning Representations*.
Mu, Jesse and Jacob Andreas. 2020. Compositional explanations of neurons.
Nie, Yixin, 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, Online.
Pang, Deric, Lucy H. Lin, and Noah A. Smith. 2019. Improving natural language inference with a pretrained parser.
Poliak, Adam, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis Only Baselines in Natural Language Inference. In *Joint Conference on Lexical and Computational Semantics (StarSem)*.
Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.
Shah, Darsh J, Tal Schuster, and Regina Barzilay. 2020. Automatic fact-guided sentence modification. In *Proceedings of the AAAI Conference on Artificial Intelligence*.
Shorten, Connor and Taghi M Khoshgoftaar. 2019. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):60.
Tu, Lifu, Garima Lalwani, Spandana Gella, and He He. 2020. An empirical study on robustness to spurious correlations using pre-trained language models.
Utama, Prasetya Ajie, Nafise Sadat Moosavi, and Iryna Gurevych. 2020a. Mind the trade-off: Debiasing NLU models without degrading the in-distribution performance. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8717–8729, Association for Computational Linguistics, Online.
Utama, Prasetya Ajie, Nafise Sadat Moosavi, and Iryna Gurevych. 2020b. Towards debiasing NLU models from unknown biases. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7597–7610, Association for Computational Linguistics, Online.
Verma, Saurabh and Zhi-Li Zhang. 2019. Stability and generalization of graph

convolutional neural networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 1539–1548, Association for Computing Machinery, New York, NY, USA.

Wang, Alex, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in Neural Information Processing Systems*, 32. 33rd Annual Conference on Neural Information Processing Systems, NeurIPS 2019 ; Conference date: 08-12-2019 Through 14-12-2019.

Wang, Alex, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Association for Computational Linguistics, Brussels, Belgium.

Wang, Jianyu, Zhishuai Zhang, Cihang Xie, Yuyin Zhou, Vittal Premachandran, Jun Zhu, Lingxi Xie, and Alan Yuille. 2017. Visual concepts and compositional voting. *Annals of Mathematical Sciences and Applications*, 3.

Weber, Noah, Leena Shekhar, and Niranjan Balasubramanian. 2018. The fine line between linguistic generalization and failure in Seq2Seq-attention models. In *Proceedings of the Workshop on Generalization in the Age of Deep Learning*, pages 24–27, Association for Computational Linguistics, New Orleans, Louisiana.

Wendlandt, Laura, Jonathan K. Kummerfeld, and Rada Mihalcea. 2018. Factors influencing the surprising instability of word embeddings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2092–2102, Association for Computational Linguistics, New Orleans, Louisiana.

Williams, Adina, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, Association for Computational Linguistics, New Orleans, Louisiana.

Yaghoobzadeh, Yadollah, Remi Tachet, T. J. Hazen, and Alessandro Sordoni. 2019. Robust natural language inference models with example forgetting.

Zhang, Tianyi, Felix Wu, Arzoo Katiyar, Kilian Q. Weinberger, and Yoav Artzi. 2020. Revisiting few-sample bert fine-tuning.

Zhou, Xiang 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, Online.

Zhou, Xiang, Yixin Nie, Hao Tan, and Mohit Bansal. 2020. The curse of performance instability in analysis datasets: Consequences, source, and suggestions.