MITIGATING BIAS IN NATURAL LANGUAGE INFERENCE USING ADVERSARIAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recognizing the relationship between two texts is an important aspect of natural language understanding (NLU), and a variety of neural network models have been proposed for solving NLU tasks. Unfortunately, recent work showed that the datasets these models are trained on often contain biases that allow models to achieve non-trivial performance without possibly learning the relationship between the two texts. We propose a framework for building robust models by using adversarial learning to encourage models to learn latent, bias-free representations. We test our approach in a Natural Language Inference (NLI) scenario, and show that our adversarially-trained models learn robust representations that ignore known dataset-specific biases. Our experiments demonstrate that our models are more robust to new NLI datasets.

1 INTRODUCTION

Recognizing the relationship between two texts is a significant aspect of general natural language understanding (NLU) (Allen, 1995). Natural Language Inference (NLI) is often used to gauge a model's ability to understand such a relationship between two texts (Cooper et al., 1996; Dagan et al., 2006). In NLI, a model is tasked with determining whether a hypothesis (*the animal moved*) would likely be inferred from a premise (*a black cat ran*). The development of new large-scale datasets has led to a flurry of various neural network architectures for solving NLI. However, recent work has found that many NLI datasets contain biases that enable hypothesis-only models – models that are given access to the hypothesis alone – to perform surprisingly well without possibly learning the relationship between two texts. For instance, annotation artifacts and statistical irregularities in the popular Stanford Natural Language Inference dataset (SNLI) (Bowman et al., 2015) allowed hypothesis-only models to perform at double the majority class baseline, and at least 5 other recent NLI datasets contain similar biases (Gururangan et al., 2018; Poliak et al., 2018c; Tsuchiya, 2018). We will use the terms "artifacts" and "biases" interchangeably.

The existence of annotation artifacts in large-scale NLI datasets is detrimental for making progress in deep learning research for NLU. How can we trust the performance of top models if it is possible to infer the relationship without even looking at the premise? Solutions to this concern are so far unsatisfactory: constructing new datasets (Sharma et al., 2018) is costly and may still result in other artifacts; filtering "easy" examples and defining a harder subset is useful for evaluation purposes (Gururangan et al., 2018), but difficult to do on a large scale that will enable training; and compiling adversarial examples (Glockner et al., 2018) is informative but again limited by scale or diversity. Furthermore, these solutions do not address a lingering question: can we develop models that will generalize well despite many NLI datasets containing specific hypothesis-only biases?

Inspired by domain-adversarial training of neural networks (Ganin & Lempitsky, 2015; Ganin et al., 2016), we propose two architectures (Figure 1) that enable a model to perform well on other NLI datasets regardless of what annotation artifacts exist in the training corpus's hypotheses. While learning to classify the relationship between two texts, we simultaneously use adversarial learning to discourage our model from using dataset-specific biases.In this way, the resulting representations contain fewer biases, and the model is encouraged to learn the relationship between the two texts. Our experiments demonstrate that our architectures generate sentence representations that are more robust to annotation artifacts, and also transfer better: when trained on one dataset and evaluated on another, they perform better than a non-adversarial model in 9 out of 12 target datasets. The methodology can also be extended to other NLU tasks, and we outline the necessary changes to our architectures in the conclusion. To our knowledge, this is the first study that explores methods to ignore hypothesis-only biases when training NLI models.

2 Related Work

2.1 Biases and Artifacts in NLU datasets

Annotation artifacts or biases were found in multiple NLU datasets and tasks. Early work on NLI, also known as recognizing textual entailment (RTE), found biases that allowed models to perform relatively well by focusing on syntactic clues alone (Snow et al., 2006; Vanderwende & Dolan, 2006). More recently, Gururangan et al. (2018) found such artifacts in SNLI and its multi-genre extension, MNLI (Williams et al., 2018). Tsuchiya (2018) observed similar findings in SNLI and SICK, a sentence similarity dataset, and Poliak et al. (2018c) reported similar results on six datasets, including those previous three. Some of the biased annotations that were found include the use of negation words ("not", "nothing") for cases of contradiction or approximate words ("some", "various") for entailment. Poliak et al. (2018c) discussed how faulty dataset-construction methods may be responsible for these biases.

Other NLU datasets also exhibit biases. In a story cloze completion setup, Schwartz et al. (2017b) obtained a high performance by only considering the candidate endings, without even looking at the story context. In the ROC stories dataset (Mostafazadeh et al., 2016), stylistic features such as length or the use of certain words are predictive of the correct ending (Schwartz et al., 2017a; Cai et al., 2017). A similar phenomenon was observed in reading comprehension, where systems performed non-trivially well by only using the final sentence in the passage, or by ignoring the passage altogether (Kaushik & Lipton, 2018). Finally, multiple studies found non-trivial performance in visual question answering by using only the question, without any access to the image, due to biases in the question text (Zhang et al., 2016; Kafle & Kanan, 2016; Goyal et al., 2017; Kafle & Kanan, 2017; Agrawal et al., 2017).

2.2 Improving Model Robustness

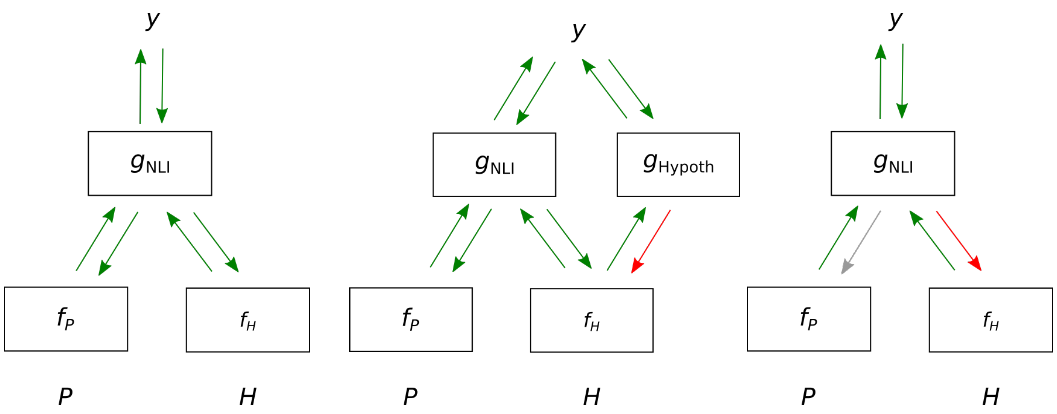
Neural networks are notoriously sensitive to adversarial examples, primarily in the machine vision field (Szegedy et al., 2014; Goodfellow et al., 2015), but also in NLP tasks like machine translation (Ebrahimi et al., 2018; Belinkov & Bisk, 2018; Heigold et al., 2018) and reading comprehension (Jia & Liang, 2017; Ribeiro et al., 2018; Mudrakarta et al., 2018). A common approach to improving robustness is to train the model on data including adversarial examples (Szegedy et al., 2014; Goodfellow et al., 2015). However, this method may not generalize well to new types of adversarial examples (Xiaoyong Yuan, 2017; Tramr et al., 2018; Belinkov & Bisk, 2018).

Domain-adversarial neural networks aim to increase robustness to domain change, by learning to be oblivious to the domain (Ganin et al., 2016). This approach requires knowledge of the domain at training time, which makes transfer learning more difficult. Our method relies on a similar idea, but we learn to ignore latent annotation artifacts. Hence, we do not require direct supervision in the form of a domain label.

Others have attempted to remove biases from learned representations. Bolukbasi et al. (2016) successful removed gender biases from word embeddings. Li et al. (2018) removed sensitive information like sex and age from text representations, and obtained improved representations especially in out-of-domain scenarios. However, other work suggests that removing such attributes from text representations may be difficult (Elazar & Goldberg, 2018). In contrast to this line of work, our final goal is not the removal of such attributes per se; instead, we strive for more robust representations that better transfer to other datasets, similar to Li et al. (2018). Very recent work has focused on applying adversarial learning to NLI. Minervini & Riedel (2018) generate adversarial examples that do not conform to logical rules and then regularize models based on those examples. Similarly, Kang et al. (2018) incorporate external linguistic resources and use a GAN-style framework to adversarially train robust NLI models. Unlike these works, we do not use external resources and we are interested in removing specific biases that allow hypothesis-only models to perform well.

3 Methodology

Let (P, H) denote a premise-hypothesis pair, and let $f : S \to \boldsymbol{v}$ denote an encoder that maps a sentence S to a vector representation $\boldsymbol{v}$, and $g : \boldsymbol{v} \to y$ a classifier that maps a vector representation $\boldsymbol{v}$ to an output label y. Our baseline NLI architecture (Figure 1a) contains the following components:



(a) Baseline NLI architecture (b) Double classifier architecture (c) Single classifier architecture

Figure 1: Illustration of (a) the baseline NLI architecture, and two proposed adversarial architectures: (b) a double-classifier adds an adversarial hypothesis-only classifier, (c) a single-classifier is trained adversarially with a random premise, and otherwise in the normal manner. Upward and downward arrows correspond to forward and backward propagation. Green or red arrows respectively mean that the gradient sign is kept as is or reversed. Gray arrow indicates that the gradient is blocked and not back-propagated.

- A **premise encoder** f_P that maps the premise P to a vector representation $\boldsymbol{p}$.
- A **hypothesis encoder** f_H that maps the hypothesis H to a vector representation $\boldsymbol{h}$.
- A **classifier** g_{NLI} that maps a premise-hypothesis vector representation to an output y.

In this model, the premise and hypothesis are each encoded with separate encoders, f_P and f_H, into $\boldsymbol{p}$ and $\boldsymbol{h}$ respectively.[1] Then their representations are combined into $[\boldsymbol{p}; \boldsymbol{h}]$[2] and fed to a classifier g_{NLI}, which predicts an output y. If f_P is not used, a model should no longer be able to successfully perform NLI. However, models without f_P achieve non-trivial results, indicating the existence of biases in datasets' hypotheses (Gururangan et al., 2018; Poliak et al., 2018c; Tsuchiya, 2018).

To overcome such biases that may limit the ability to transfer models across NLI datasets, we design two kinds of models for robust NLI, a single-classifier model and a double-classifier model. Both models aim to encourage the hypothesis representations to be free of biases via adversarial learning.

3.1 Double-classifier model

The double-classifier model, illustrated in Figure 1b, is similar to our baseline model but includes an **adversarial classifier** g_{Hypoth} that maps the hypothesis representation $\boldsymbol{h}$ to an output y. The crucial aspect of this model is the interaction between the NLI classifier, g_{NLI}, and the hypothesis classifier, g_{Hypoth}. The NLI classifier is trained to minimize the following objective:

$$L_{\text{NLI}} = L(g_{\text{NLI}}([f_P(P); f_H(H)], y) \tag{1}$$

where $L(\tilde{y}, y)$ is the cross-entropy loss. In the forward step, we feed the premise and hypothesis to their respective encoders, f_P and f_H, and forward their joint representation $[\boldsymbol{p}; \boldsymbol{h}]$ to the classifier. In the backward step, gradients are back-propagated from the classifier into the premise and hypothesis encoders, f_P and f_H, in the normal fashion.

In contrast, the hypothesis classifier g_{Hypoth} is adversary with respect to the input. It is also trained to minimize the cross-entropy loss, but its forward/backward propagation is different. In the forward step, we feed the hypothesis into its encoder f_H and forward its representation $\boldsymbol{h}$ to the classifier. In the backward step, we first back-propagate gradients through the classifier, as before. However,

[1] This is common to many NLI models (Rocktäschel et al., 2016; Mou et al., 2016; Cheng et al., 2016; Nie & Bansal, 2017; Chen et al., 2017), although some share information between the encoders via attention (Parikh et al., 2016; Duan et al., 2018). Extending our approach to this scenario would require some caution.

[2] For simplicity, we assume here that the two are concatenated. In the experiments we consider a more sophisticated method to represent sentence pairs.

before back-propagating to the encoder, we reverse the gradients (Ganin & Lempitsky, 2015). This simple step aims to discourage the model from learning patterns that may be useful for classification when considering only the hypothesis. The adversary minimizes the following objective:

$$L_{\text{Adv}} = L(g_{\text{Hypoth}}(\text{GRL}_\lambda(f_H(H)), y) \tag{2}$$

where GRL_λ is a gradient reversal layer. To control the interplay between g_{NLI} and g_{Hypoth} we set two hyper-parameters: λ_{Loss} controls the importance of the adversarial loss function, and λ_{Enc} controls the weight of the adversarial update by multiplying the gradients after reversing them. This is implemented by the scaled gradient reversal layer, GRL_λ. The final loss function is defined by:

$$L = L_{\text{NLI}} + \lambda_{\text{Enc}} L_{\text{Adv}} \tag{3}$$

Limitation and a cryptographic perspective The double-classifier model is conceptually simple. However, it has a potential limitation because of the separate adversarial classifier. In theory, it is possible that the adversary and the hypothesis encoder will co-adapt during training, such that the hypothesis representation still contains biased information but the adversary cannot utilize it. Thus we may be fooled to think that the biases were removed, while in fact they are encoded in a way that is accessible to the normal classifier but not to the adversary. A similar situation arises in neural cryptography (Abadi & Andersen, 2016), where an encryptor Alice and a decryptor Bob communicate while an adversary Eve tries to eavesdrop on their communication. Alice and Bob are analogous to the hypothesis encoder f_H and the normal classifier g_{NLI}, while Eve is analogous to the adversary g_{Hypoth}. Secret communication is analogous to solving NLI without using biases.

In their asymmetric encryption experiments, Abadi & Andersen observed seemingly secret communication, which on closer look the adversary was able to eavesdrop on. Here, if the adversarial classifier does not perform well, we might be tricked into thinking that the encoded representation does not contain any biases, while in fact they are still hidden in the representation. Our next architecture aims to prevent such a situation from possibly happening by folding the NLI and adversarial classifiers into a single network.

3.2 Single-classifier model

The single-classifier model also includes the premise and hypothesis encoders, f_P and f_H. However, it only has one classifier, g, which acts as both a normal NLI classifier and an adversarial hypothesis classifier. Having only one classifier aims at reducing the risk that a separate adversary would give a false impression of success, learning to not use hidden biases in the hypothesis representations. Since g must also do well on NLI with normal training, it is less likely to ignore hidden biases.

To achieve this, we consider two modes of operation. In the normal mode, we get a premise-hypothesis pair from the training data, feed them through their encoders, forward to the classifier, and back-propagate in the normal fashion. In the adversarial mode, we get a premise-hypothesis from the training data and replace the premise with a *random* one. In the forward step, we feed the new pair to the encoders and classifier as before, predicting the entailment decision corresponding to the original premise-hypothesis pair. In the backward step, we first back-propagate through the classifier as usual. Then, we block the premise encoder and only back-propagate to the hypothesis encoder. In this case, we reverse the gradients going into the hypothesis encoder, as in the double-classifier. This procedure is shown in Figure 1c. The adversarial loss function is defined as:

$$L_{\text{RandAdv}} = L(g_{\text{NLI}}([\text{GRL}_0(f_P(P_i)); \text{GRL}_\lambda(f_H(H_i))], y) \tag{4}$$

Here GRL_0 implements gradient blocking on the premise encoder by using the identity function in the forward step and a zero gradient during the backward step. At the same time, GRL_λ reverses the gradient going into the hypothesis encoder and scales it by λ_{Enc}, as before.

The single-classifier model has the advantage of a simpler architecture, avoiding the need to train two different classifiers, which may result in failed adversarial learning as described before. However, it has a more complicated training regime, in choosing random examples. In practice, we set a hyper-parameter $\lambda_{\text{Rand}} \in [0, 1]$ that controls what fraction of the examples are random. The final loss function combines the two operation modes with a random variable $z \sim \text{Bernoulli}(\lambda_{\text{Rand}})$:

$$L = (1 - z)L_{\text{NLI}} + zL_{\text{RandAdv}} \tag{5}$$

Table 1: Accuracies of the non-adversarial baseline and double/single classifiers models, trained on SNLI and evaluated on target datasets. Δ = differences between baseline and adversarial models.

Target Dataset	Baseline	Double	Δ	Single	Δ
SCITAIL	58.14	57.67	-0.47	51.08	-7.06
ADD-ONE-RTE	66.15	66.15	0	83.46	17.31
JOCI	41.5	41.74	0.24	39.63	-1.87
MPE	57.65	58.1	0.45	52.35	-5.3
DPR	49.86	50.96	1.1	49.41	-0.45
MNLI matched	45.86	47.24	1.38	43.76	-2.1
FN+	50.87	52.48	1.61	57.03	6.16
MNLI mismatched	47.57	49.24	1.67	43.66	-3.91
SICK	25.64	27.44	1.8	56.75	31.11
GLUE	38.5	40.49	1.99	43.21	4.71
SPR	52.48	58.99	6.51	65.43	12.94
SNLI hard	68.02	66.27	-1.75	55.6	-12.42

4 Experimental Setup

Data To determine how well our proposed architectures enable a model to perform well on NLI datasets despite the high presence of the annotation artifacts in the training corpus, we use a total of 11 NLI datasets – the 10 datasets that Poliak et al. (2018c) investigated in their hypothesis-only study plus GLUE's diagnostic test set that was carefully constructed to not contain hypothesis-biases (Wang et al., 2018). The most popular recent NLI datasets are arguably the Stanford Natural Language Inference (SNLI) (Bowman et al., 2015) dataset and its successor, the Multi-genre Natural Language Inference (MNLI) (Williams et al., 2018) dataset. These datasets are *human elicited*: they were created by having human generate a corresponding hypothesis for a given premise and NLI label. Since SNLI is known to contain significant annotation artifacts/biases (Gururangan et al., 2018; Poliak et al., 2018c), we will demonstrate the robustness of our methods by training our adversarial models on SNLI, and evaluating on all other datasets. We also evaluate on SNLI-hard, a subset of the test set that is thought to contain fewer biases (Gururangan et al., 2018).

The second category of NLI datasets we consider are *human-judged* datasets that used automatic methods to pair context and hypothesis sentences and then relied on humans to label the pairs: Scitail (Khot et al., 2018), ADD-ONE-RTE (Pavlick & Callison-Burch, 2016), Johns Hopkins Ordinal Commonsense Inference (JOCI) (Zhang et al., 2017), Multiple Premise Entailment (MPE) (Lai et al., 2017), and Sentences Involving Compositional Knowledge (SICK) (Marelli et al., 2014). We also consider datasets that are automatically *recast* from existing NLU datasets into NLI. We use the three datasets recast by White et al. (2017) to evaluate a number of semantic phenomena: FrameNet+ (FN+) (Pavlick et al., 2015), Definite Pronoun Resolution (DPR) (Rahman & Ng, 2012), Semantic Proto-Roles (SPR) (Reisinger et al., 2015).

As many of these target datasets have different label spaces than SNLI, we define a mapping from our models' predictions to each target dataset's labels. These mappings are available in Appendix A.1.

Implementation Details We adopt `InferSent`'s method (Conneau et al., 2017) for learning sentence representations from NLI data as our basic NLI architecture. In `InferSent`, each sentence is encoded by a bidirectional long short-term memory (LSTM) network whose hidden states are max-pooled. The premise and hypothesis representations are combined via a method introduced by Mou et al. (2016) and passed to a one hidden layer neural network.[3] We chose this model because it works well and its architecture is representative of many NLI models. However, our methodology can be applied to other models as well. We follow the `InferSent` training regime, using SGD with an initial learning rate of 0.1. See Appendix B.1 for hyper-parameter settings and more details.

5 Results

Table 1 reports the results of our proposed architectures compared to the non-adversarial baseline model. In each case, we tune the hyper-parameters on each target dataset's development set and

[3] Specifically, representations are concatenated, subtracted, and multiplied element-wise.

evaluate on the corresponding test set.[4] The double-classifier model outperforms the baseline in 9 of the 12 target datasets ($\Delta > 0$), though most of the improvements are small. The single-classifier only outperforms the baseline in 5 datasets, 4 of which are cases where the double-classifier also outperformed the baseline. These gains are much larger than the gains of the double-classifier. The fact that the two architectures agree to a large extent on which datasets benefit from adversarial training is a validation of our basic approach. As our results improve on the target datasets, we note that the double classifier models' performance on SNLI does not drastically decrease, even when the improvement on the target dataset is large (for example, the SPR case). For these models, the performance on SNLI drops by just an average of 1.11 (0.65 STDV). For the single-classifier, there is a large decrease on SNLI for many of the adversarial models as the models drop by an average of 11.19 (12.71 STDV). For these models, when we see large improvement on a target dataset, we often see a large drop on SNLI. For example, on ADD-ONE-RTE, the single classifier outperforms the baseline by roughly 17% but performs almost 50% lower than the baseline on SNLI.[5]

A priori, we expect the adversarial models to benefit most when there are either no biases or biases that differ from ones in the training data. Indeed, both our adversarial architectures obtain improved results on the GLUE diagnostic test set, which was designed to be bias-free. We do not see improvements on the SNLI hard subset (Gururangan et al., 2018), indicating that it may still have biases not identified by the authors, a possibility they also acknowledge.

To estimate the amount of bias that differ in the datasets, we compare the hypothesis-only results from Poliak et al. (2018c) with a hypothesis-only model trained on SNLI and tested on the target datasets. Since the results drop significantly below the majority class baseline (MAJ) on all but one dataset (Figure 4, Appendix C.2), we believe that these target NLI datasets contain different biases than those in SNLI. The largest difference is on SPR where the hypothesis-only model trained on SNLI performs over 50% worse than when trained on SPR. On MNLI, this hypothesis-only model performs 10% above MAJ, compared to the roughly 20% when trained on MNLI, suggesting that MNLI contains similar biases as SNLI. This may explain why our adversarial models only slightly outperform our baseline on MNLI. The hypothesis-only model of Poliak et al. (2018c) did not outperform MAJ on DPR, ADD-ONE-RTE, SICK, and MPE. We observe improvements with adversarially trained models that are tested on all these datasets, to varying degrees (from 0.45 on MPE to 31.11 on SICK). On the other hand, we also see improvements on datasets with biases (high performance of hypothesis-only model), most noticeably on SPR. The only exception seems to be SCITAIL, where we do not improve although it has different biases than SNLI. However, when we strengthen the adversary (in the analysis below), the double-classifier outperforms the baseline. Our results demonstrates that our approach is robust to many datasets with different types of biases.

Fine-tuning on target datasets Our main goal is to determine whether adversarial learning can allow a model to perform well across multiple datasets by ignoring dataset-specific artifacts. In turn, we did not update the models' parameters on other datasets. However, what if we are given different amounts of training data for a new NLI dataset? Is our adversarial approach still helpful? To answer these questions, we updated four existing models, on increasing sizes of training data from different target datasets (namely, MNLI and SICK). The four models are (1) our baseline model trained on SNLI, (2) an adversarial double classifier trained on SNLI, (3) an adversarial single classifier trained on SNLI, and (4) our baseline model trained on the target dataset (MNLI/SICK). Both MNLI and SICK have the same label spaces as SNLI, allowing us to hold that variable constant. We use SICK because our adversarial models achieved good gains on it (Table 1). We also use MNLI, even though we saw small gains there, because MNLI's training set is large, allowing us to consider a wide range of different training set sizes.[6]

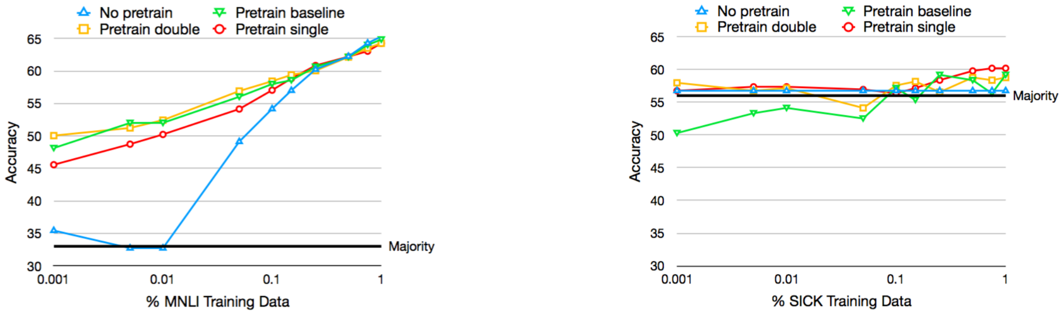
Figure 2 shows the results on the dev sets. In MNLI, there is little to no gain from adversarial pre-training compared to non-adversarial pre-training. This is expected, as we saw relatively small gains with the adversarial models, and can be explained by SNLI and MNLI having similar biases. In SICK, adversarial pre-training is better in most data regimes. The single-classifier is especially helpful, as it is the first to beat the model without pre-training (after using 25% of the training data).[7]

[4] For MNLI, since the test sets are not available, we tune on the matched dev set and evaluate on the mismatched dev set, or vice versa. For GLUE, we tune on MNLI matched.

[5] See Table 4 in Appendix C for the corresponding results on SNLI for each of the results listed in Table 1.

[6] We discuss necessary implementation details for these experiments in Appendix B.2

[7] Note that SICK is a small dataset (10K training examples), which explains why the model without pre-training does not benefit from more data, as it just achieves the majority class performance.



(a) MNLI (b) SICK

Figure 2: Effect of pre-training with adversarially trained models.

Table 2: Results with a stronger adversary in the double-classifier vs. a non-adversarial baseline.

Dataset	Baseline	Double	Δ	Dataset	Baseline	Double	Δ
SNLI hard	68.02	63.81	-4.21	SCITAIL	58.14	60.82	2.68
JOCI	41.5	39.29	-2.21	ADD-ONE	66.15	68.99	2.84
SNLI	84.22	82.4	-1.82	GLUE	38.5	41.58	3.08
DPR	49.86	49.41	-0.45	FN+	50.87	56.31	5.44
MNLI matched	45.86	46.12	0.26	SPR	52.48	58.68	6.2
MNLI mismatched	47.57	48.19	0.62	SICK	25.64	36.59	10.95
MPE	57.65	58.6	0.95				

6 Analysis

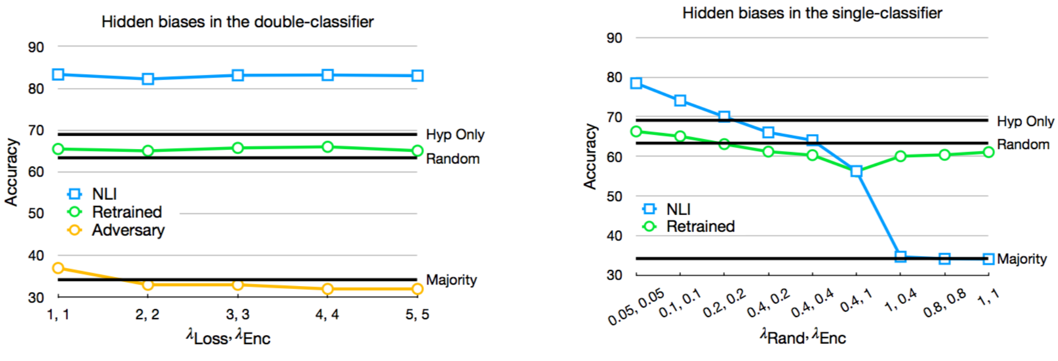
Stronger adversary Does a stronger adversary improve the quality of the transferred representations? Intuitively, we expect more adversarial training to hurt performance on the original data, where biases are useful, but improve on target data, which may have different or no biases. To test this, we trained adversarial models with larger hyper-parameter values on the adversary (see details in Appendix B.1). While there are other ways to make adversaries stronger, such as increasing the number or size of hidden layers (Elazar & Goldberg, 2018), we are especially interested in the effect of these hyper-parameters as they control the trade-off between normal and adversarial training.

Table 2 shows the results of double-classifier models with a stronger adversary. As expected, performance on SNLI test sets decreases more, but many of the other datasets benefit from a stronger adversary (compared with Table 1). As for the single-classifier model, we found large drops in quality even in our basic configurations (Appendix C.3), so we do not increase its strength further.

Hidden biases in the representation Recall our motivation for the single-classifier model: we were concerned that the learned hypothesis representations may seem to be bias-free (low adversarial performance), while in fact there are still biases hidden in them. In the cryptographic analogy, it might appear that Alice and Bob communicate secretly (akin to solving NLI without biases), while in fact their communication can be decrypted (has biases). Indeed, Abadi & Andersen (2016) found that, “upon resetting and retraining Eve, the retrained adversary was able to decrypt messages nearly as well as Bob was”. We perform an analogous experiment here: given a trained adversarial model, we freeze the hypothesis encoder f_H and retrain a new, hypothesis-only classifier. We evaluate its quality to determine whether the (frozen) hypothesis representations have hidden biases.

Figure 3 shows the results on SNLI’s dev set. A few trends can be noticed. First, we confirm that, in the double-classifier case (Figure 3a), the adversary is indeed trained to perform poorly on the task (orange line), while the normal NLI classifier (blue line) performs much better. However, as suspected, retraining a classifier on frozen hypothesis representations (green line) leads to improved performance. In fact, the retrained classifier performs close to the fully trained hypothesis-only baseline from (Poliak et al., 2018c), indicating that the hypothesis representation still contains biases.

Interestingly, we found that even a frozen random encoder captures biases in the hypothesis, as a classifier trained on it performs fairly well (63.26%), and far above the majority baseline (34.28%). One reason might be that just the word embeddings (which are pre-trained) contain significant information that propagates even through a random encoder. Others have also found that random encodings contain non-trivial information (Conneau et al., 2018; Zhang & Bowman, 2018). Relatedly,



(a) Double-classifier

(b) Single-classifier

Figure 3: Results when retraining a classifier on a frozen hypothesis encoder compared to the adversarial NLI training.

Table 3: Indicator words and the % decrease in the correlation of double (D) and single (S) models with the "contradiction" label compared to the non-adversarial baseline.

Word	D(1,1)	S(0.4,1)	S(1,1)	Word	D(1,1)	S(0.4,1)	S(1,1)
sleeping	15.63	53.13	-81.25	driving	-8.33	50	-66.67
Nobody	14.29	42.86	14.29	alone	0	83.33	0
cat	7.14	57.14	-85.71	asleep	-18.75	50	12.5
no	0	52.94	-52.94	empty	-16.67	83.33	-16.67
eats	37.5	87.5	-25	naked	0	83.33	-33.33

Elazar & Goldberg found that adversarially trained text classification models still have demographic attributes in their hidden representations despite efforts to remove them.

Indicator words Certain words in SNLI are more correlated with specific entailment labels than others, especially with negation words ("not", "nobody", "no") correlated with *contradiction* (Gururangan et al., 2018; Poliak et al., 2018c). Here we investigate whether our adversarial models make predictions that are less impacted by such biases. For each of the most biased contradiction words in SNLI, we computed the probability that an adversarial model (or the baseline model) predicts an example as contradiction, given that the hypothesis has the word. Table 3 shows the top 5 examples in the training set (Appendix C.4 has more examples). The single-classifier with $\lambda_{\text{Rand}} = 0.4$, $\lambda_{\text{Enc}} = 1$ predicts contradiction much less frequently than the baseline on examples with these words. This configuration was the strongest adversarial model that still performed reasonably well on SNLI (Appendix C.3). With these hyper-parameters, the single-classifier appears to remove some of the biases learned by the baseline. We also provide two other adversarial configurations that do not show such an effect, to illustrate that this behavior highly depends on the adversarial hyper-parameters.

7 Conclusion and Future Work

Biases in annotations are a major source of concern for the quality of NLI datasets and systems. In this paper, we presented a solution for combating annotation biases based on adversarial learning. We designed two architectures that discourage the hypothesis encoder from learning the biases, and instead obtain a more unbiased representation. We empirically evaluated our approach in a transfer learning scenario, where we found our models to perform better than a non-adversarial baseline on a range of datasets. We also investigated what biases remain in the latent representations.

The methodology developed in this work can be extended to deal with biases in other NLU tasks, where one is concerned with finding the relationship between two objects. For example, in Reading Comprehension, a question is being asked about a passage; in story cloze completion, an ending is judged with respect to a context; and in Visual Question Answering, a question is asked about an image. In all these cases, the second element (question, ending, and question, respectively) may contain biases. Our adversarial architectures naturally apply to any model that relies on encoding this biased element, and may help remove such biases from the latent representation. We hope to encourage such investigation in the broader research community.

REFERENCES

Martn Abadi and David G. Andersen. Learning to Protect Communications with Adversarial Neural Cryptography. *arXiv*, 2016. URL https://arxiv.org/abs/1610.06918.

Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. Don't Just Assume; Look and Answer: Overcoming Priors for Visual Question Answering. *arXiv preprint arXiv:1712.00377*, 2017.

James Allen. *Natural language understanding*. Pearson, 1995.

Yonatan Belinkov and Yonatan Bisk. Synthetic and natural noise both break neural machine translation. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=BJ8vJebC-.

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in Neural Information Processing Systems*, pp. 4349–4357, 2016.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2015.

Zheng Cai, Lifu Tu, and Kevin Gimpel. Pay attention to the ending:strong neural baselines for the roc story cloze task. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 616–622. Association for Computational Linguistics, 2017. doi: 10.18653/v1/P17-2097. URL http://www.aclweb.org/anthology/P17-2097.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. Enhanced lstm for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1657–1668. Association for Computational Linguistics, 2017. doi: 10.18653/v1/P17-1152. URL http://www.aclweb.org/anthology/P17-1152.

Jianpeng Cheng, Li Dong, and Mirella Lapata. Long short-term memory-networks for machine reading. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 551–561, Austin, Texas, November 2016. Association for Computational Linguistics. URL https://aclweb.org/anthology/D16-1053.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 670–680. Association for Computational Linguistics, 2017. URL http://aclweb.org/anthology/D17-1070.

Alexis Conneau, Germán Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2126–2136. Association for Computational Linguistics, 2018. URL http://aclweb.org/anthology/P18-1198.

Robin Cooper, Dick Crouch, Jan Van Eijck, Chris Fox, Johan Van Genabith, Jan Jaspars, Hans Kamp, David Milward, Manfred Pinkal, Massimo Poesio, and Steven Pullman. Using the framework. Technical Report LRE 62-051 D-16, The FraCaS Consortium, 1996.

Ido Dagan, Oren Glickman, and Bernardo Magnini. The pascal recognising textual entailment challenge. In *Machine learning challenges. evaluating predictive uncertainty, visual object classification, and recognising tectual entailment*, pp. 177–190. Springer, 2006.

Chaoqun Duan, Lei Cui, Xinchi Chen, Furu Wei, Conghui Zhu, and Tiejun Zhao. Attention-fused deep matching network for natural language inference. IJCAI 2018, July 2018. URL https://www.microsoft.com/en-us/research/publication/attention-fused-deep-matching-network-natural-language-inference/.

Javid Ebrahimi, Daniel Lowd, and Dejing Dou. On adversarial examples for character-level neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 653–663. Association for Computational Linguistics, 2018. URL http://aclweb.org/anthology/C18-1055.

Yanai Elazar and Yoav Goldberg. Adversarial Removal of Demographic Attributes from Text Data. *arXiv preprint arXiv:1808.06640*, 2018.

Yaroslav Ganin and Victor Lempitsky. Unsupervised Domain Adaptation by Backpropagation. In *International Conference on Machine Learning*, pp. 1180–1189, 2015.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

Max Glockner, Vered Shwartz, and Yoav Goldberg. Breaking nli systems with sentences that require simple lexical inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 650–655, Melbourne, Australia, July 2018. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P18-2103.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples. In *International Conference on Learning Representations (ICLR)*, 2015.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *CVPR*, volume 1, pp. 3, 2017.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 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)*, pp. 107–112, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N18-2017.

Georg Heigold, Günter Neumann, and Josef van Genabith. How Robust Are Character-Based Word Embeddings in Tagging and MT Against Wrod Scramlbing or Randdm Nouse? In *Proceedings of the 13th Conference of The Association for Machine Translation in the Americas (Volume 1: Research Track*, pp. 68–79, March 2018.

Robin Jia and Percy Liang. Adversarial Examples for Evaluating Reading Comprehension Systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2011–2021, Copenhagen, Denmark, September 2017.

K. Kafle and C. Kanan. Answer-Type Prediction for Visual Question Answering. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4976–4984, June 2016. doi: 10.1109/CVPR.2016.538.

Kushal Kafle and Christopher Kanan. Visual question answering: Datasets, algorithms, and future challenges. *Computer Vision and Image Understanding*, 163:3–20, 2017.

Dongyeop Kang, Tushar Khot, Ashish Sabharwal, and Eduard Hovy. Adventure: Adversarial training for textual entailment with knowledge-guided examples. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2418–2428, Melbourne, Australia, July 2018. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P18-1225.

Divyansh Kaushik and Zachary C Lipton. How much reading does reading comprehension require? a critical investigation of popular benchmarks. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

Tushar Khot, Ashish Sabharwal, and Peter Clark. SciTail: A textual entailment dataset from science question answering. In *AAAI*, 2018.

Alice Lai, Yonatan Bisk, and Julia Hockenmaier. Natural language inference from multiple premises. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 100–109, Taipei, Taiwan, November 2017. Asian Federation of Natural Language Processing. URL http://www.aclweb.org/anthology/I17-1011.

Yitong Li, Timothy Baldwin, and Trevor Cohn. Towards Robust and Privacy-preserving Text Representations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 25–30. Association for Computational Linguistics, 2018. URL http://aclweb.org/anthology/P18-2005.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella bernardi, and Roberto Zamparelli. A sick cure for the evaluation of compositional distributional semantic models. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis (eds.), *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pp. 216–223, Reykjavik, Iceland, May 2014. European Language Resources Association (ELRA). ISBN 978-2-9517408-8-4. URL http://www.lrec-conf.org/proceedings/lrec2014/pdf/363_Paper.pdf. ACL Anthology Identifier: L14-1314.

Pasquale Minervini and Sebastian Riedel. Adversarially regularising neural nli models to integrate logical background knowledge. In *Proceedings of the 22nd Conference on Computational Natural Language Learning (CoNLL 2018)*. Association for Computational Linguistics, 2018.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 839–849, San Diego, California, June 2016. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N16-1098.

Lili Mou, Rui Men, Ge Li, Yan Xu, Lu Zhang, Rui Yan, and Zhi Jin. Natural language inference by tree-based convolution and heuristic matching. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 130–136, Berlin, Germany, August 2016. Association for Computational Linguistics. URL http://anthology.aclweb.org/P16-2022.

Pramod Kaushik Mudrakarta, Ankur Taly, Mukund Sundararajan, and Kedar Dhamdhere. Did the model understand the question? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1896–1906. Association for Computational Linguistics, 2018. URL http://aclweb.org/anthology/P18-1176.

Yixin Nie and Mohit Bansal. Shortcut-stacked sentence encoders for multi-domain inference. In *Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP*, pp. 41–45, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W17-5308.

Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2249–2255. Association for Computational Linguistics, 2016. doi: 10.18653/v1/D16-1244. URL http://www.aclweb.org/anthology/D16-1244.

Ellie Pavlick and Chris Callison-Burch. Most "babies" are "little" and most "problems" are "huge": Compositional entailment in adjective-nouns. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2164–2173. Association for Computational Linguistics, 2016. doi: 10.18653/v1/P16-1204. URL `http://www.aclweb.org/anthology/P16-1204`.

Ellie Pavlick, Travis Wolfe, Pushpendre Rastogi, Chris Callison-Burch, Mark Dredze, and Benjamin Van Durme. Framenet+: Fast paraphrastic tripling of framenet. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 408–413, Beijing, China, July 2015. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P15-2067`.

Adam Poliak, Yonatan Belinkov, James Glass, and Benjamin Van Durme. On the evaluation of semantic phenomena in neural machine translation using natural language inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 513–523, New Orleans, Louisiana, June 2018a. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/N18-2082`.

Adam Poliak, Aparajita Haldar, Rachel Rudinger, J. Edward Hu, Ellie Pavlick, Aaron Steven White, and Benjamin Van Durme. Collecting Diverse Natural Language Inference Problems for Sentence Representation Evaluation. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2018b.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pp. 180–191, New Orleans, Louisiana, June 2018c. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/S18-2023`.

Altaf Rahman and Vincent Ng. Resolving complex cases of definite pronouns: The winograd schema challenge. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 777–789, Jeju Island, Korea, July 2012. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/D12-1071`.

Drew Reisinger, Rachel Rudinger, Francis Ferraro, Craig Harman, Kyle Rawlins, and Benjamin Van Durme. Semantic proto-roles. *Transactions of the Association for Computational Linguistics*, 3: 475–488, 2015. ISSN 2307-387X. URL `https://tacl2013.cs.columbia.edu/ojs/index.php/tacl/article/view/674`.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Semantically Equivalent Adversarial Rules for Debugging NLP models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 856–865. Association for Computational Linguistics, 2018. URL `http://aclweb.org/anthology/P18-1079`.

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomas Kocisky, and Phil Blunsom. Reasoning about entailment with neural attention. In *International Conference on Learning Representations (ICLR)*, 2016.

Roy Schwartz, Maarten Sap, Ioannis Konstas, Leila Zilles, Yejin Choi, and Noah A. Smith. The Effect of Different Writing Tasks on Linguistic Style: A Case Study of the ROC Story Cloze Task. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pp. 15–25. Association for Computational Linguistics, 2017a. doi: 10.18653/v1/K17-1004. URL `http://www.aclweb.org/anthology/K17-1004`.

Roy Schwartz, Maarten Sap, Ioannis Konstas, Leila Zilles, Yejin Choi, and Noah A. Smith. Story cloze task: UW NLP system. In *Proceedings of LSDSem*, 2017b.

Rishi Sharma, James Allen, Omid Bakhshandeh, and Nasrin Mostafazadeh. Tackling the story ending biases in the story cloze test. In *Proceedings of the 56th Annual Meeting of the Association* *for Computational Linguistics (Volume 2: Short Papers)*, pp. 752–757, Melbourne, Australia, July 2018. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P18-2119.

Rion Snow, Lucy Vanderwende, and Arul Menezes. Effectively using syntax for recognizing false entailment. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, 2006. URL http://www.aclweb.org/anthology/N06-1005.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014.

Florian Tramr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rkZvSe-RZ.

Masatoshi Tsuchiya. Performance Impact Caused by Hidden Bias of Training Data for Recognizing Textual Entailment. In *11th International Conference on Language Resources and Evaluation (LREC2018)*, 2018.

Lucy Vanderwende and William B Dolan. What syntax can contribute in the entailment task. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pp. 205–216. Springer, 2006.

Alex Wang, Amapreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

Aaron Steven White, Pushpendre Rastogi, Kevin Duh, and Benjamin Van Durme. Inference is everything: Recasting semantic resources into a unified evaluation framework. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 996–1005, Taipei, Taiwan, November 2017. Asian Federation of Natural Language Processing. URL http://www.aclweb.org/anthology/I17-1100.

Adina Williams, Nikita Nangia, and Samuel Bowman. 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)*, pp. 1112–1122. Association for Computational Linguistics, 2018. URL http://aclweb.org/anthology/N18-1101.

Qile Zhu Xiaolin Li Xiaoyong Yuan, Pan He. Adversarial Examples: Attacks and Defenses for Deep Learning. *arXiv preprint arXiv:1712.07107*, 2017.

Kelly Zhang and Samuel Bowman. Language Modeling Teaches You More than Translation Does: Lessons Learned Through Auxiliary Task Analysis. In *EMNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, 2018.

P. Zhang, Y. Goyal, D. Summers-Stay, D. Batra, and D. Parikh. Yin and Yang: Balancing and Answering Binary Visual Questions. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5014–5022, June 2016. doi: 10.1109/CVPR.2016.542.

Sheng Zhang, Rachel Rudinger, Kevin Duh, and Benjamin Van Durme. Ordinal common-sense inference. *Transactions of the Association for Computational Linguistics*, 5:379–395, 2017. ISSN 2307-387X. URL https://transacl.org/ojs/index.php/tacl/article/view/1082.

A Data details

A.1 Mapping labels

Each premise-hypothesis pair in SNLI is labeled as either ENTAILMENT, NEUTRAL, or CONTRADICTION. MNLI, SICK, JOCI, and MPE use the same label space. Since examples in SciTail are labeled as ENTAILMENT or NEUTRAL, when evaluating on SciTail, we convert the model's CONTRADICTION to NEUTRAL. ADD-ONE-RTE and the recast datasets also model NLI as a binary prediction task. However, their label sets are ENTAILED and NOT-ENTAILED. In these cases, when the models predict ENTAILMENT, we convert the example to ENTAILED, and when the models predict NEUTRAL or CONTRADICTION, we map the label to NOT-ENTAILED.

B Training Details

B.1 Adversarial Training Details

We tune the adversarial hyper-parameters $\lambda_{\text{Loss}}, \lambda_{\text{Enc}}, \lambda_{\text{Rand}}$ by sweeping in the range $\{0.05, 0.1, 0.2, 0.4, 0.8, 1.0\}$. For the stronger adversary experiments (Table 2) we consider the range $\{1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0\}$. In each dataset, we choose the best-performing model on the development set and report its quality on the test set.

We follow the `InferSent` training regime, using SGD with an initial learning rate of 0.1. See Conneau et al. (2017) for details.

B.2 Updating Training

One drawback of using MNLI is that gold labels for the test set are not publicly available. We remove 10K examples from the training set and treat them as our development set. Here, we use the MNLI matched development set as our test set since we assume that the new dataset, that we would like our robust NLI models to generalize well on, contains consistent domains and genres.

C Additional Results

C.1 Performance on SNLI with adversarial training

Table 4 shows the results of transferring representations to new datsets. These results complement Table 1. In each dataset, we tune the adversarial hyper-parameters on the dev set and report test set accuracies. We also give the performance on the SNLI test set.

Table 4: Results of transferring representations to new datasets. Scores are accuracies on the corresponding dataset. Left block: test results on target datasets. Right block: test results on SNLI with the models that performed best on each target dataset. Δ are differences from the non-adversarial baseline. In all cases the models are trained on SNLI.

	On Target Dataset				On SNLI			
Target Dataset	Double	Δ	Single	Δ	Double	Δ	Single	Δ
SCITAIL	57.67	-0.47	51.08	-7.06	84.04	-0.18	75.16	-9.06
ADD-ONE-RTE	66.15	0	83.46	17.31	81.93	-2.29	34.59	-49.63
JOCI	41.74	0.24	39.63	-1.87	83.78	-0.44	78.3	-5.92
MPE	58.1	0.45	52.35	-5.3	83.65	-0.57	83.68	-0.54
DPR	50.96	1.1	49.41	-0.45	83.49	-0.73	76.41	-7.81
MNLI matched	47.24	1.38	43.76	-2.1	82.97	-1.25	75.29	-8.93
FN+	52.48	1.61	57.03	6.16	82.28	-1.94	83.78	-0.44
MNLI mismatched	49.24	1.67	43.66	-3.91	82.97	-1.25	75.29	-8.93
SICK	27.44	1.8	56.75	31.11	83.65	-0.57	75.29	-8.93
GLUE	40.49	1.99	43.21	4.71	82.97	-1.25	75.29	-8.93
SPR	58.99	6.51	65.43	12.94	82.46	-1.76	70.21	-14.01
SNLI					83.56	-0.66	78.3	-5.92
SNLI hard	66.27	-1.75	55.6	-12.42				

C.2 Hypothesis-only baselines

Figure 4 shows the results of several baselines in multiple datasets: the majority baseline and the hypothesis-only baselines trained on either SNLI or each dataset's training set, and evaluated on each target dataset's test set. Notice that the hypothesis-only model trained on SNLI drops in performance when evaluated on other models. When the hypothesis-only model trained on SNLI is tested on the target datasets, the model performs below the majority baseline (except for MNLI), indicating that the biases in SNLI's hypotheses do not occur in the other datasets' hypotheses. This is not surprising since these datasets contain different types of biases owing to noisy generation methods (FN+) (Poliak et al., 2018a), social and cognitive biases (SNLI & MNLI) (Poliak et al., 2018c), or even "non-uniform distributions from original dataset that have been recast" (SPR) (Poliak et al., 2018b).

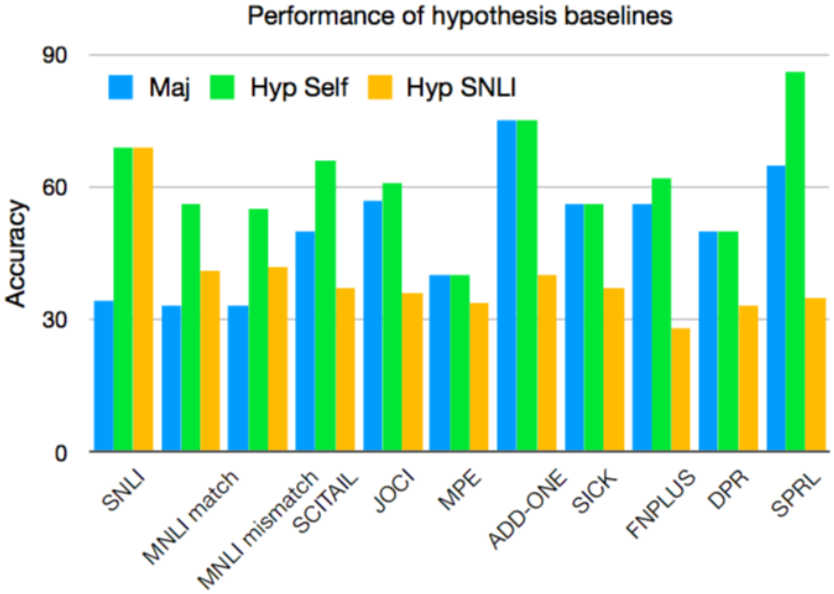
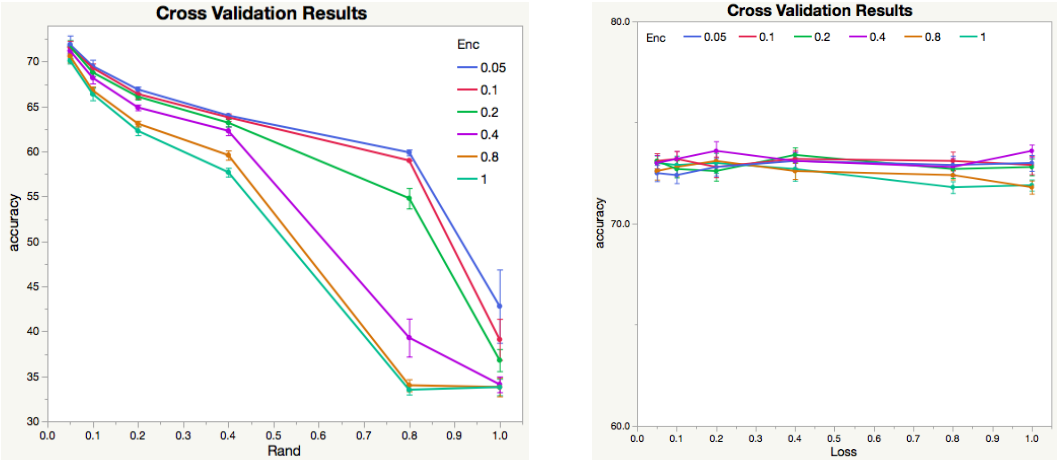


Figure 4: Majority (Maj) and hypothesis-only baselines, trained on each dataset (Hyp Self) or on SNLI (Hyp SNLI) baselines. The differences determine whether each dataset has different types of biases in the hypothesis than the biases in SNLI's hypotheses.

C.3 Hyper-parameter sweeps

Here we provide cross-validation results with different settings of our hyper-parameters. Figure 5a shows the dev set results with different configurations of the single classifier. Notice that performance degrades quickly when we increase the fraction of random premises (large λ_{Rand}). In contrast, the results with the double classifier (Figure 5b) are more stable.



(a) Single classifier

(b) Double classifier

Figure 5: Cross-validation results.

C.4 Indicator words

Table 5 shows the top 20 indicator words in SNLI, to complement the discussion in Section 6.

Table 5: Indicator words and how correlated they are with "contradiction" predictions.

		Score		Percentage difference from baseline		
Word	Freq	Gold	Baseline	Double (1,1)	Single (0.4,1)	Single (1,1)
sleeping	108	0.88	0.24	15.63	53.13	-81.25
driving	53	0.81	0.32	-8.33	50	-66.67
Nobody	52	1	0.42	14.29	42.86	14.29
alone	50	0.9	0.32	0	83.33	0
cat	49	0.84	0.31	7.14	57.14	-85.71
asleep	43	0.91	0.39	-18.75	50	12.5
no	31	0.84	0.36	0	52.94	-52.94
empty	28	0.93	0.3	-16.67	83.33	-16.67
eats	24	0.83	0.3	37.5	87.5	-25
naked	20	0.95	0.46	0	83.33	-33.33
sleeps	20	0.95	0.25	0	50	-100
television	17	0.88	0.14	-100	0	-300
moon	14	0.93	0.33	-50	0	0
sleep	12	1	0.18	0	50	-250
tv	12	0.92	0.33	0	25	-100
pizza	12	0.92	0.27	-33.33	33.33	-33.33
nothing	11	1	0.29	0	0	-50
single	11	0.82	0.57	0	75	0
No	9	0.89	0.5	16.67	50	0
anything	8	1	1	50	83.33	83.33