HypoNLI: Exploring the Artificial Patterns of Hypothesis-only Bias in Natural Language Inference

Tianyu Liu\textsuperscript{1}, Xin Zheng\textsuperscript{3}, Baobao Chang\textsuperscript{1,2}, Zhifang Sui\textsuperscript{1,2}
\textsuperscript{1}MOE Key Lab of Computational Linguistics, School of EECS, Peking University
\textsuperscript{2}Peng Cheng Laboratory, Shenzhen, China \textsuperscript{3}Beijing University of Posts and Telecommunications
tianyu0421@pku.edu.cn, zheng_xin@bupt.edu.cn, cbb@pku.edu.cn, szf@pku.edu.cn

Abstract

Many recent studies have shown that for models trained on datasets for natural language inference (NLI), it is possible to make correct predictions by merely looking at the hypothesis while completely ignoring the premise. In this work, we manage to derive adversarial examples in terms of the hypothesis-only bias and explore eligible ways to mitigate such bias. Specifically, we extract various phrases from the hypotheses (artificial patterns) in the training sets, and show that they have been strong indicators to the specific labels. We then figure out ‘hard’ and ‘easy’ instances from the original test sets whose labels are opposite to or consistent with those indications. We also set up baselines including both pretrained models (BERT, RoBERTa, XLNet) and competitive non-pretrained models (InferSent, DAM, ESIM). Apart from the benchmark and baselines, we also investigate two debiasing approaches which exploit the artificial pattern modeling to mitigate such hypothesis-only bias: down-sampling and adversarial training. We believe those methods can be treated as competitive baselines in NLI debiasing tasks.

Keywords: Natural Language Inference, Hypothesis-only Bias, Artificial Patterns

1. Introduction

Natural language inference (NLI) (also known as recognizing textual entailment) is a widely studied task which aims to infer the relationship (e.g., entailment, contradiction, neutral) between two fragments of text, known as premise and hypothesis (Dagan et al., 2006; Dagan et al., 2013). NLI models are usually required to determine whether a hypothesis is true (entailment) or false (contradiction) given the premise, or whether the truth value cannot be inferred (neutral). A proper NLI decision should apparently rely on both the premise and the hypothesis. However, some recent studies (Gururangan et al., 2018; Poliak et al., 2018; Tsuchiya, 2018) have shown that it is possible for a trained model to identify the true label by only looking at the hypothesis without observing the premise. The phenomenon is referred to as annotation artifacts (Gururangan et al., 2018), statistical irregularities (Poliak et al., 2018) or partial-input heuristics (Feng et al., 2019). In this paper we use the term hypothesis-only bias (Poliak et al., 2018) to refer to this phenomenon.

Such hypothesis-only bias originates from the human annotation process of data collection. In the data collection process of many large-scale NLI datasets such as SNLI (Bowman et al., 2015) and MultiNLI (Williams et al., 2018), human annotators are required to write new sentences (hypotheses) based on the given premise and a specified label among entailment, contradiction and neutral. Some of the human-elicited hypotheses contain patterns that spuriously correlate to some specific labels. For example, 85.2\% of the hypothesis sentences which contain the phrase video games were labeled as contradiction. The appearance of video games in hypothesis can be seen as a stronger artificial indicator to the label contradiction.

To get a deeper understanding of the specific bias captured by NLI models in the training procedure, we try to extract explicit surface patterns from the training sets of SNLI and MultiNLI, and show that the model can easily get decent classification accuracy by merely looking at these patterns. After that, we derive hard (adversarial) and easy subsets from the original test sets. They are derived based on the indication of the artificial patterns in the hypotheses. The gold labels of easy subsets are consistent with such indication while those of hard subsets are opposite to such indication. The model performance gap on easy and hard subsets shows to what extend a model can mitigate the hypothesis-only bias.

After analyzing some competitive NLI models, including both non-pretrained models like InferSent (Conneau et al., 2017), DAM (Parikh et al., 2016) and ESIM (Chen et al., 2017), we find that the hypothesis-only bias makes NLI models vulnerable to the adversarial (hard) instances which are against such bias (accuracy < 60\% on InferSent), while these models get much higher accuracy (accuracy > 95\% on InferSent) on the easy instances. This is an evidence to show that the NLI models might be over-estimated as they benefit a lot from the hints of artificial patterns.

A straightforward way is to eliminate these human artifacts in the human annotation process, such as encouraging human annotators to use more diverse expressions or do dataset adversarial filtering (Zellers et al., 2018) and multi-round annotation (Nie et al., 2019b). However in this way, the annotation process would inevitably become more time-consuming and expensive.

To this end, this paper explores two ways based on the derived artificial patterns to alleviate the hypothesis-only bias in the training process: down-sampling and adversarial debiasing. We hope they would serve as competitive baselines for other NLI debiasing methods. Down-sampling aims at removing the hypothesis-only bias in NLI training sets by removing those instances in which the correct labels may easily be revealed by artificial patterns. Furthermore, we exploring adversarial debiasing methods (Belinkov et al., 2019; Belinkov et al., 2018) for the sentence vector-based models in NLI (Yang et al., 2016; Lin et al., 2017; Wu et
## Artificial Pattern Collection

‘Pattern’ in this work refers to (maybe nonconsecutive) word segments in the hypothesis sentences. We try to identify the ‘artificial patterns’ which spuriously correlate to a specific label due to certain human artifacts. We use $H(M, t, \lambda)$ to represent a set of artificial patterns. $M$ and $t$ denotes the max length of the pattern and the max distance between two consecutive words in a pattern, respectively. For a artificial pattern $b \in H(M, t, \lambda)$, there exists a specific label $l$ for $b$ that the conditional probability $p(l | b) = \frac{|count(b,l)|}{|count(b)|} > \lambda$. For example, for the underlined pattern ‘A # # outside’ in Table 1, the length of this pattern is 3, and the distance between the consecutive words ‘A’ and ‘outside’ is 2. Its conditional probability with the label *entailment* is 95.9%. Notably, all the recognized artificial patterns in our paper appear in at least 50 instances of the training sets to avoid misrecognition.

In the rest of paper, unless otherwise specified, we set $M = 3$, $t = 3$. By doing so, we only tune the hyper-parameter $\lambda$ in $H(3,3,\lambda)$ to decide using a rather strict (smaller $\lambda$) or mild (bigger $\lambda$) strategy while deriving the artificial patterns.

### Analysis of Hypothesis-only Bias

Previous work (Gururangan et al., 2018; Poliak et al., 2018) trained a sentence-based hypothesis-only classifier which achieves decent accuracy. Different from them, we show that in Table 2 the classifier which merely uses the artificial patterns as features achieves comparable performance with the fasttext (Joulin et al., 2016) classifier. Table 2 shows the classifier based on multi-word patterns with the default $M$ and $t$ (see Footnote 2) achieves much higher accuracy than that based on only unigram patterns ($H(1,1,0.5)$).

We also compare the test accuracies on the easy and hard sets (Sec 2.3) of the baseline models (I-9,D-9,E-9) in Table 5 and 6. Empirically we find that the NLI models achieve very high accuracy on the easy sets while performing poorly on the hard sets. We also observe the same tendency in the models trained on the randomly downsampled training sets (e.g. I-1, I-3, I-5, I-7, etc.). It shows that NLI models fit the artificial patterns in the training set very well, which makes them fragile to the adversarial examples (hard set) which are against these patterns. Thus we assume the artificial patterns contributes to the hypothesis-only bias.

### Hard and Easy Subsets

Some instances contain artificial patterns that are strong indicators to the specific labels. We treat the instances in the test sets which are consistent with such indication as ‘easy’ ones and those instances which are against such indication as ‘hard’ ones.

For easy subsets, the labels of all the artificial patterns in

---

**Table 1**: Top 3 artificial patterns sorted by the pattern-label conditional probability $p(l | b)$ (Sec 2.1). The listed patterns appear at least in 500/200 instances in SNLI/MultiNLI training sets, notably the numbers 500/200 here are chosen only for better visualization. ‘#’ is the placeholder for an arbitrary token. The underlined artificial pattern serves as an example in Sec 2.1.

| Pattern          | Entailment | Neutral | Contradiction |
|------------------|------------|---------|---------------|
| A human          | 96.4       | 99.7    | 99.8          |
| A # # outdoors. | 95.9       | 94.1    | 96.2          |
| A # # outside.  | 89.8       | 88.6    | 94.2          |
| is near # # .   | 87.6       | 87.0    | 93.5          |

**Table 2**: The accuracies of the hypothesis-only classifiers on SNLI test and MultiNLI dev sets. We train a MLP classifier with unigrams (Unigram) or multi-words patterns (Pattern) as features. Details in Sec 2.2.

| Model | SNLI | MultiNLI |
|-------|------|----------|
|       | Matched | Mismatched |
| majority class | 34.3 | 35.4 | 35.2 |
| fasttext | 67.2 | 53.7 | 52.5 |
| Unigram | 60.2 | 49.8 | 49.6 |
| Pattern | 64.4 | 52.9 | 52.9 |
and hard subsets. Hard subsets should be close to each other. Besides we should ensure the size of derived subsets. For the sake of simplicity, we utilize \( \lambda = 0 \) and \( \lambda = 0.7 \) as the thresholds to derive easy and hard subsets for SNLI and MultiNLI respectively (Sec 2.1. and Footnote 2). For the hard subsets, on the other side, the indications of the artificial patterns should be all different from gold labels. We also show a hard instance below: in this situation, the artificial patterns ‘no chest hair’ and ‘his way’ may misguide the NLI models to the wrong answers. Notably we do not put instances with conflicting indications (e.g. an instance with 2 artificial patterns, one of which has the same label with the gold label while the other does not) into easy or hard subsets to build more challenging adversarial examples.

The sizes of hard and easy sets actually depend on how we harvest artificial pattern, i.e. \( \lambda \) in \( H(M = 3, t = 3, \lambda) \) (Sec 2.1. and Footnote 2). For the sake of simplicity, we utilize \( \lambda = 0.8 \) and \( \lambda = 0.7 \) as the thresholds to derive easy and hard subsets for SNLI and MultiNLI respectively in the following experiments, as adopting a relatively bigger \( \lambda \) can choose the instances which largely accord with the artificial patterns and are thus eligible to serve as adversarial examples.

The sizes of easy and hard sets in SNLI test set, MultiNLI-matched dev set and MultiNLI-mismatched dev set are 327/1760; 410/1032; 371/1085 respectively. The performance of an ideally unbiased NLI model on the easy and hard sets should be close to each other. Besides we should not see huge gap between the model accuracy on the easy and hard subsets.

| Premise | Hypothesis | Gold Label | Artificial patterns |
|---------|------------|------------|---------------------|
| The dogs are on the bed | The dogs are on the bed | CONTRADICTION | (bed, CONTRADICTION, 83.2%) |
| | | | (The dogs are \# on, CONTRADICTION, 82.9%) |

(a) An easy instance

The specific hypotheses must be consistent with the gold labels. We show an easy instance below: the artificial patterns ‘The dogs are \# on’ and ‘bed’ (bed is the last word of the sentence) are strong indicators to the correct classification. For the hard subsets, on the other side, the indications of the artificial patterns should be all different from gold labels. We also show a hard instance below: in this situation, the artificial patterns ‘no chest hair’ and ‘his way’ may misguide the NLI models to the wrong answers.

Table 3: Examples for easy and hard instances. The indications of artificial patterns are consistent with the gold label in the easy case while they are against the gold label in the hard case. The triple \( (P, l, p) \) show the related label indications \( l \) for specific artificial patterns \( P \) and their conditional probability \( p \).

| Premise | Hypothesis | Gold Label | Artificial patterns |
|---------|------------|------------|---------------------|
| A bare-chested man fitting his head and arm into a toilet seat ring while spectators watch in a city. | A gentleman with no chest hair, wrangles his way through a toilet seat. | ENTAILMENT | (no, CONTRADICTION, 82.7%) |
| | | | (his way, NEUTRAL, 82.4%) |

(b) A hard instance

Table 3: Examples for easy and hard instances. The indications of artificial patterns are consistent with the gold label in the easy case while they are against the gold label in the hard case. The triple \( (P, l, p) \) show the related label indications \( l \) for specific artificial patterns \( P \) and their conditional probability \( p \).

Notably we do not put instances with conflicting indications (e.g. an instance with 2 artificial patterns, one of which has the same label with the gold label while the other does not) into easy or hard subsets to build more challenging adversarial examples.

The sizes of hard and easy sets actually depend on how we harvest artificial pattern, i.e. \( \lambda \) in \( H(M = 3, t = 3, \lambda) \) (Sec 2.1. and Footnote 2). For the sake of simplicity, we utilize \( \lambda = 0.8 \) and \( \lambda = 0.7 \) as the thresholds to derive easy and hard subsets for SNLI and MultiNLI respectively in the following experiments, as adopting a relatively bigger \( \lambda \) can choose the instances which largely accord with the artificial patterns and are thus eligible to serve as adversarial examples.

The sizes of easy and hard sets in SNLI test set, MultiNLI-matched dev set and MultiNLI-mismatched dev set are 327/1760; 410/1032; 371/1085 respectively. The performance of an ideally unbiased NLI model on the easy and hard sets should be close to each other. Besides we should not see huge gap between the model accuracy on the easy and hard subsets.

| Premise | Hypothesis | Gold Label | Artificial patterns |
|---------|------------|------------|---------------------|
| The dogs are # on | The dogs are on the bed | CONTRADICTION | (bed, CONTRADICTION, 83.2%) |
| | | | (The dogs are \# on, CONTRADICTION, 82.9%) |

(a) An easy instance

Table 3: Examples for easy and hard instances. The indications of artificial patterns are consistent with the gold label in the easy case while they are against the gold label in the hard case. The triple \( (P, l, p) \) show the related label indications \( l \) for specific artificial patterns \( P \) and their conditional probability \( p \).

| Premise | Hypothesis | Gold Label | Artificial patterns |
|---------|------------|------------|---------------------|
| A bare-chested man fitting his head and arm into a toilet seat ring while spectators watch in a city. | A gentleman with no chest hair, wrangles his way through a toilet seat. | ENTAILMENT | (no, CONTRADICTION, 82.7%) |
| | | | (his way, NEUTRAL, 82.4%) |

(b) A hard instance

Table 3: Examples for easy and hard instances. The indications of artificial patterns are consistent with the gold label in the easy case while they are against the gold label in the hard case. The triple \( (P, l, p) \) show the related label indications \( l \) for specific artificial patterns \( P \) and their conditional probability \( p \).

Table 4: Model baselines on the proposed hard and easy test sets. For MultiNLI, we trained the models using matched dev sets as the valid set and reported the test accuracies on mismatched dev sets. ‘UW+CMU’ refers to the adversarial set (Sec 2.4.) detected by a neural based hypothesis-only classifier.

2.4. Baselines

We set up both pretrained and non-pretrained model baselines for the proposed adversarial datasets. We rerun their public available codebase with the default hyper-parameter and optimizer settings, including InferSent\(^6\), DAM\(^7\), ESIM\(^8\) and RoBERTa (base)\(^9\) for BERT, XLNet and RoBERTa, we concatenate the premise sentence and hypothesis sentence with [SEP] token as the input. For output classifier, we use a linear mapping to transform the vector at the position of [CLS] token at the last layer of these pretrained models to a normalized 3-element vector (using softmax) which represents the scores for each label. We report the test accuracies on easy, hard subsets and the UW+CMU hard subsets\(^10\). For Table 4 we can tell that the proposed hard sets are more challenging than UW+CMU hard subsets.

3. Exploring Debiasing Methods

3.1. Down-sampling Baselines

Sec 2.2. verifies that the artificial patterns lead to correct hypothesis-only classification, which motivates us to remove such patterns in the training sets by down-sampling. Specifically we down-sample the training sets of SNLI and MultiNLI and retrain 3 prevailing NLI models: InferSent, DAM and ESIM.

3.2. Artificial Pattern-Only Baselines

"Artificial pattern-only" is a baseline that uses artificial patterns and their labels to generate an input for NLI models. The objective is to see if NLI models are sensitive to the presence of artificial patterns and learn to disambiguate them from actual evidence in the input. In the down-sampling baselines, the artificial patterns are not considered. However, in the artificial pattern-only baselines, the artificial patterns are considered as evidence, even if they are not consistent with the gold label.

| Model | Full | Easy | Hard | UW+CMU |
|-------|------|------|------|--------|
| InferSent | 84.5 | 97.2 | 58.9 | 69.3 |
| DAM | 85.8 | 97.8 | 62.1 | 72.0 |
| ESIM | 87.6 | 97.7 | 68.2 | 75.2 |
| BERT | 90.5 | 98.2 | 71.2 | 80.3 |
| XLNet | 90.9 | 98.0 | 73.6 | 80.7 |
| RoBERTa | 91.7 | 98.9 | 75.8 | 82.7 |

(a) Models trained on SNLI

| Model | Full | Easy | Hard |
|-------|------|------|------|
| InferSent | 70.4 | 92.7 | 54.4 |
| DAM | 70.5 | 92.0 | 55.1 |
| ESIM | 76.7 | 93.9 | 65.6 |
| BERT | 83.4 | 95.2 | 75.0 |
| XLNet | 86.5 | 96.3 | 78.2 |
| RoBERTa | 87.2 | 96.5 | 81.4 |

(b) Models trained on MultiNLI

---

\(^3\)"no" is different from ‘No’ shown in Table 1 as the latter indicates the word appears in the beginning of the sentence.

\(^4\)MultiNLI’s pattern-label conditional possibilities are generally smaller than those of SNLI as shown in Table 1. So we use smaller \( \lambda \) to ensure the size of derived subsets.

\(^5\)The datasets used in this paper can be found in https://tyliupku.github.io/publications/

\(^6\)https://github.com/facebookresearch/InferSent

\(^7\)https://github.com/harvardnlp/decomp-attn

\(^8\)https://github.com/coetaur0/ESIM

\(^9\)https://github.com/huggingface/transformers
one artificial pattern. When adopting smaller \( \lambda \) training instance as a biased one even if it contains only \( \lambda \) is exactly the same.

Choosing down-sampling threshold

We down-sampled the training sets by removing the biased instances ('Debias') and randomly down-sampling ('Rand'), the latter has the same training size and label distribution with with 'Debias' mode for a fair comparison. The downsampled training sizes are 4.0%, 19.8%, 43.8%, 67.4% and 100% of the whole training size (392702) for \( \lambda \in \{0.5, 0.6, 0.7, 0.8, 1.0\} \) respectively. Note that we do not report the scores on MultiNLI test sets as they are unable to access. The bold numbers mark higher scores between 'Rand' and 'Debias' mode for each \( \lambda \). The underlined numbers highlight the highest scores in each column.

### 3.1.1. Downsampling Details

We down-sampled the training sets by removing the biased instances ('Debias' mode) that contain the artificial patterns.

Choosing down-sampling threshold \( \lambda \): The threshold \( \lambda \) is exactly the same \( \lambda \) defined in Sec 2.1. We consider a training instance as a biased one even if it contains only one artificial pattern. When adopting smaller \( \lambda \), we harvest more artificial patterns as described in Sec 2.1. Accordingly more training instances would be treated as biased ones and then filtered. In a word, smaller \( \lambda \) represents more strict down-sampling strategy in terms of filtering the artificial patterns. \( \lambda = 0.5 \) serves as the lower bound because

Table 5: Model performance on the SNLI test set. We report the average scores of multiple independent runs. \( \Delta_{\text{Hard}}^{\text{Easy}} \) represents the gap between hard and easy test sets (lower is better). \( \lambda \) is the debiasing threshold. We use 2 ‘modes’ to down-sample the training sets, namely biased instances removing ('Debias') and randomly down-sampling ('Rand'), the latter has the same training size and label distribution with with 'Debias' mode for a fair comparison. The downsampled training sizes are 4.0%, 19.8%, 43.8%, 67.4% and 100% of the whole training size (549867) for \( \lambda \in \{0.5, 0.6, 0.7, 0.8, 1.0\} \) respectively. Note that when \( \lambda = 1.0 \), we use the whole training set without any downsampling.

| \( \lambda \) | No. | Mode | Full | Easy | Hard | \( \Delta_{\text{Hard}}^{\text{Easy}} \) |
|---|---|---|---|---|---|---|
| 0.5 | I-1 | Rand | 76.4 | 93.8 | 48.7 | 45.1 |
| 0.5 | I-2 | Debias | 66.9 | 64.6 | 56.0 | 8.6 |
| 0.6 | I-3 | Rand | 81.1 | 96.1 | 54.1 | 42.0 |
| 0.6 | I-4 | Debias | 76.9 | 79.8 | 58.0 | 21.8 |
| 0.7 | I-5 | Rand | 82.8 | 96.9 | 56.0 | 40.9 |
| 0.7 | I-6 | Debias | 80.9 | 86.4 | 61.5 | 24.9 |
| 0.8 | I-7 | Rand | 83.5 | 96.9 | 56.6 | 40.3 |
| 0.8 | I-8 | Debias | 82.9 | 90.4 | 60.0 | 30.4 |
| 1.0 | I-9 | All | 84.5 | 97.2 | 58.9 | 38.3 |

(a) InterSent trained on SNLI

| \( \lambda \) | No. | Mode | Full | Easy | Hard | \( \Delta_{\text{Hard}}^{\text{Easy}} \) |
|---|---|---|---|---|---|---|
| 0.5 | D-1 | Rand | 74.1 | 92.3 | 46.7 | 45.6 |
| 0.5 | D-2 | Debias | 67.5 | 67.8 | 53.3 | 14.5 |
| 0.6 | D-3 | Rand | 82.4 | 96.3 | 56.7 | 39.6 |
| 0.6 | D-4 | Debias | 79.3 | 84.4 | 62.1 | 21.3 |
| 0.7 | D-5 | Rand | 84.4 | 97.3 | 59.4 | 37.9 |
| 0.7 | D-6 | Debias | 83.0 | 89.6 | 63.1 | 26.5 |
| 0.8 | D-7 | Rand | 85.3 | 97.8 | 60.1 | 37.7 |
| 0.8 | D-8 | Debias | 84.6 | 93.5 | 62.6 | 30.9 |
| 1.0 | D-9 | All | 85.8 | 97.8 | 62.1 | 35.7 |

(b) DAM trained on SNLI

| \( \lambda \) | No. | Mode | Full | Easy | Hard | \( \Delta_{\text{Hard}}^{\text{Easy}} \) |
|---|---|---|---|---|---|---|
| 0.5 | E-1 | Rand | 76.8 | 94.6 | 48.9 | 45.7 |
| 0.5 | E-2 | Debias | 65.3 | 62.5 | 53.8 | 7.7 |
| 0.6 | E-3 | Rand | 83.6 | 96.6 | 62.2 | 34.4 |
| 0.6 | E-4 | Debias | 78.6 | 79.4 | 63.6 | 15.8 |
| 0.7 | E-5 | Rand | 85.9 | 97.2 | 64.2 | 35.0 |
| 0.7 | E-6 | Debias | 83.8 | 88.2 | 68.8 | 19.4 |
| 0.8 | E-7 | Rand | 86.9 | 97.3 | 67.9 | 29.4 |
| 0.8 | E-8 | Debias | 86.2 | 92.1 | 70.9 | 21.3 |
| 1.0 | E-9 | All | 87.6 | 97.6 | 68.2 | 29.4 |

(c) ESIM trained on SNLI
ple $n_3$ instances for each pre-defined label from training set.

**Convincing scores of multiple runs:** To relieve the randomness of randomly down-sampling and model initialization, for the ‘rand’ mode in Table 5 and 6, we firstly randomly down-sample the training set (with the label distribution balanced) according to different $\lambda$ for 5 times and get 5 randomly down-sampled training sets for each $\lambda$. Then for each down-sampled training set, we run 3 independent experiments with random model initialization under the same experimental settings. So each score in the ‘rand’ mode of Table 5 and 6 comes from 15 independent runs. The scores in the ‘debias’ mode of Table 5 and 6 come from 5 independent runs with random model initialization.

### 3.1.2. Discussions

From table 5 and 6, we observe that: 1) The NLI models fit the bias patterns in the hypotheses very well even in the small-scale randomly down-sampled training sets (I-1, D-1 and E1) which only accounts for 4.0% of the original training set (SNLI), as the performance gaps between easy and hard subsets in these settings are still huge (>40% for SNLI in Table 5). 

2) Under the same $\lambda$, the proposed ‘debias’ down-sampling not only outperforms its ‘rand’ counterpart in terms of hard subsets, but also greatly reduce the performance gap on easy and hard sets.

3) The gains on hard sets on MultiNLI are smaller than those on SNLI as MultiNLI is less biased regarding the pattern-label conditional probability (Table 1). Down-sampling achieves larger gains on more biased datasets. In SNLI, the ‘debias’ down-sampling even outperforms the baseline models (I-8 vs I-9, D-8 vs D-9, E-8 vs E-9), which is only 67.5% of the baseline models.

(Gururangan et al., 2018) expressed concerns upon down-sampling (DS) methods: 1) Will removing the artificial patterns cause new artifacts? (e.g. removing the word ‘no’, which is a strong indicator for contradiction may leave the remaining dataset with this word mostly appearing in the neutral or entailment classes thus create new artifact) and 2) Will DS methods prevent the models to learn specific inference phenomena (e.g. ‘animal’ is a hypernym of ‘dog’)?

First of all, different from (Gururangan et al., 2018) which only considered unigram patterns, our artificial patterns are mostly multi-word patterns rather than unigram patterns as the former usually has larger concurrent probability $p(l|b)$ as shown in Table 1. Our intention is to use the multi-word patterns to capture the specific ways of expression (human artifacts), rather than single words, of the human annotators. For the first concern, instead of filtering the unigram ‘no’, we prefer removing multi-word patterns which contain ‘no’, such as ‘There are no’ or ‘no # on’ for MultiNLI as shown in Table 1. For the hypernym mentioned in the second concern, as we prefer filtering multi-word patterns like ‘The dogs are # on’, we would not deliberately filter the unigram ‘dog’ unless adopting very aggressive DS strategy ($\lambda = 0.5$) in both SNLI and MultiNLI.

---

**Figure 1:** The illustration of the sentence-level debiasing framework, which is elaborated in Sec 3.2.1.

### 3.2. Adversarial Debiasing

Since the hypothesis-only bias comes solely from the hypothesis sentence, we wonder if it is possible to get rid of these biases via debiasing the hypothesis sentence vector. More specifically, we focus on the ‘sentence vector-based models’ category as defined on SNLI’s web page. Notably the idea of debiasing NLI via adversarial training has been proposed before (Belinkov et al., 2019; Belinkov et al., 2018). We hereby briefly introduce how we implement our adversarial training and how we incorporate instance reweighting method in this framework. In the following experiments, we use the full training sets without any down-sampling. We use the InferSent (Conneau et al., 2017) (BiLSTM with max pooling) model as the benchmark sentence encoder.

### 3.2.1. Adversarial Debiasing Framework

As shown in Fig 1 given the outputs $s_h = E_h(x_h)$, $s_p = E_p(x_s)$ of hypothesis and premise encoders $E_h$, $E_p$, we are interested in predicting the NLI label $y$ using a classifier $C$, $p_C(y|s_h = E_h(x_h), s_p = E_p(x_s))$. In addition, we train a hypothesis-only discriminator trying to predict the correct label $y$ solely from the hypothesis sentence representation $s_h$ by modeling $p_D(y|s_h = E_h(x_h))$. We formulate the training process in the adversarial setting by a min-max game. Specifically we train the discriminator $D$ to predict the label using only hypothesis sentence vector. Additionally we train the sentence encoder $E_h$, $E_p$ and the classifier $C$ to fool the discriminator $D$ without hurting inference ability. $\gamma$ is a hyper-parameter which controls the degree of debiasing.

$$
\min_{E_h, E_p, C} \max_D \mathbb{E}_{x_h, x_s, y \sim p(X,Y)} \log p_D(y|E_h(x_h)) - \log p_C(y|E_h(x_h), E_p(x_s))
$$

(1)

We train the encoders, discriminator and classifier in Eq (1) together with a gradient reversal layer (Ganin et al., 2016) as shown in Fig 1. We negate the gradients from the discriminator $D$ (red arrow in Fig 1) to push the hypothesis encoder $E_h$ to the opposite direction while update its parameters. The usage of gradient reversal layer makes it easier to optimize the min-max game in Eq (1).

---

10 It would be more challenging to manipulate the gradients in the non-sentence vector-based models, e.g. models which contain interactions between hypothesis and premise sentence encoders like (Chen et al., 2017a). We leave this to the future work.

11 https://nlp.stanford.edu/projects/snli/
Chen et al. [2018] than training the two adversarial components alternately like Generative Adversarial Nets (GANs) (Goodfellow et al. [2014]). We update the model parameters \( \theta \) by gradient descending (\( m \) is the batch size):

\[
\theta_{D_{\text{new}}}^{\text{old}} = \theta_{D_{\text{old}}} - \frac{1}{m} \sum_{i=1}^{m} \nabla_{\theta_{D}} \log p_{D_{\text{new}}}(y_{i}'\mid E_h(x_{q_{i}}))
\]

\[
\theta_{C_{\text{new}}}^{\text{old}} = \theta_{C_{\text{old}}} - \frac{1}{m} \sum_{i=1}^{m} \nabla_{\theta_{C}} \log p_{C_{\text{new}}}(y_{i}'\mid E_h(x_{q_{i}}), E_p(x_{p_{i}}))
\]

\[
\theta_{E_{h_{\text{new}}}}^{\text{old}} = \theta_{E_{h_{\text{old}}}} - \frac{1}{m} \sum_{i=1}^{m} \nabla_{\theta_{E_{h}}} \log p_{C_{\text{new}}}(y_{i}'\mid E_h(x_{q_{i}}), E_p(x_{p_{i}})) + \frac{\gamma}{m} \sum_{i=1}^{m} \nabla_{\theta_{E_{h}}} \log p_{D_{\text{new}}}(y_{i}'\mid E_h(x_{q_{i}}))
\]

\[\text{gradient reverse}\]

3.2.2. Guidance from Artificial Patterns

The artificial patterns turns out to be useful guidances for both the discriminator \( D \) and the classifier \( C \) as they indicate whether an instance is biased or not. We thus reweight the training instances in the training set based on the division of ‘biased’ and ‘unbiased’ training subsets.

**Guidance for Discriminator:** During the adversarial process, we optimize the discriminator \( D \) by maximizing the log likelihood loss like Eq\[2\]. We find increasing the weights of the biased instances in the training set is of great help to the adversarial debiasing model. Because in this way, the discriminator can learn more from the biased instances to better fit the hypothesis-only bias. The whole adversarial debiasing training process could benefit from a stronger hypothesis-only discriminator. Formally, we replace negative log likelihood loss function in Eq[2] with a weighted loss function:

\[
\frac{1}{m} \sum_{i=1}^{m} \mathbb{1}\{ (x_{i}, y_{i}) \in D_{\text{unbias}} \} \log p_{D}(y_{i}'\mid E_h(x_{q_{i}})) + \alpha_{1} \mathbb{1}\{ (x_{i}, y_{i}) \in D_{\text{bias}} \} \log p_{D}(y_{i}'\mid E_h(x_{q_{i}}))
\]

\[(5)\]

, where \( D = D_{\text{unbias}} \cup D_{\text{bias}} \) denotes the whole training corpus. The division of biased and unbiased training subsets depends on the debiasing threshold \( \lambda \) (just like the down-sampling threshold in Table[3] and [6]). \( \alpha_{1} \geq 1 \) is a hyper-parameter which reflects the attention on biased instances for the hypothesis-only discriminator.

**Guidance for Classifier:** Similar to the re-weighting method in Eq[5], we also apply the re-weighting strategy on the parameter learning for the inference classifier in Eq[3]. We hope the classifier can capture the concrete semantics in NLI instead of over-fitting the artificial patterns in the hypotheses. Thus we increase the weights of the unbiased training subset in the loss function of Eq[3]:

\[
\frac{1}{m} \sum_{i=1}^{m} \mathbb{1}\{ (x_{i}, y_{i}) \in D_{\text{bias}} \} \log p_{C}(y_{i}'\mid E_h(x_{q_{i}}), E_p(x_{p_{i}})) + \alpha_{2} \mathbb{1}\{ (x_{i}, y_{i}) \in D_{\text{unbias}} \} \log p_{C}(y_{i}'\mid E_h(x_{q_{i}}), E_p(x_{p_{i}}))
\]

\[(6)\]

, where \( \alpha_{2} \geq 1 \) is a threshold to control the attention the models pay on the unbiased instances.

3.2.3. Training Details

Apart from the weighted loss functions guided by the artificial patterns, we also investigate the following two techniques in the adversarial training process.

**Multiple discriminators:** The min-max game in Eq[1] could benefit from stronger discriminators. So we try \( k \in \{1, 2, 3\} \) discriminators to enhance its ability to do hypothesis-only classifications. In our experiments, we find that \( k = 2 \) is the best configuration for the discriminator.

**Dynamic reweighting:** For hyper-parameter \( \alpha(\alpha_{1} \text{ and } \alpha_{2}) \) in Eq[5] and Eq[6] respectively, we find it useful to adjust \( \alpha \) dynamically in the training process. \( \alpha_{0} \) and \( \alpha_{i} \) represents the initial values we set before training and its value after \( t \) training iterations respectively. Additionally we set up a hyper-parameter \( \phi \) to control the gap of model accuracies \( \delta \) on the easy and hard subsets in the dev set.

\[
\alpha^{t+1} = \begin{cases} 
\max(\alpha^{t} + \epsilon, \alpha^{0}), & \delta \geq \phi \\
\max(\alpha^{t} - \epsilon, \alpha^{0}), & \delta < \phi
\end{cases}
\]

\[(7)\]

where \( \epsilon \) is a hyper-parameter set as 0.5 for models trained on both datasets. Besides, we set \( \phi \) as 0.15 and 0.10 for SNLI and MultiNLI respectively. Notably although we update the hyper-parameters \( \alpha_{1} \) and \( \alpha_{2} \) dynamically in different iterations based on \( \phi \), we still select the model which has the best performance on the dev sets as the best model in each run.

**Parameter settings:** We use grid search to find the best hyper-parameter settings: \( \alpha_{1}, \alpha_{2} \in \{1, 3, 5, 10\}, \gamma \in \{0.5, 1, 3, 5\} \) in Eq[5] and Eq[6]. We also try \( \lambda \in \{0.5, 0.6, 0.7, 0.8\} \) as the threshold to split \( D_{\text{bias}} \) and \( D_{\text{unbias}} \) in Eq[5] and [6]. Specifically, we treat the instances which contain the artificial patterns in H(3,3,\lambda) (Sec[2.1]).

Footnote[2] as \( D_{\text{bias}} \), and the remaining instances as \( D_{\text{unbias}} \). For the results in Table[7] we set \( \gamma = 3 \) and \( \lambda = 0.7 \) as the threshold for separating the biased and unbiased subsets in Eq[5] and [6]. For a fair comparison, we do not tune any hyper-parameter in the InferSent encoder, the learning rate and the optimizer setting. The results of ‘dInferSent’ and its variations in Table[7] comes from 5 independent runs with random initialization.

3.2.4. Discussions

From Table[7] we observe that although the performance gap between the easy and hard subsets is reduced to some extent by the vanilla dInferSent models in both SNLI and MultiNLI. The model still does not reach our expectation to lower the gap between hard and easy sets. We assume this is because the denoising discriminator in Fig[1] somewhat impedes the inference ability of the NLI models as it may disturb the hypothesis sentence encoder especially when the sentences do not contain hypothesis-only bias. The explicit guidance (‘+Guidance’) from the artificial patterns alleviates this issue in both datasets as in this way the discriminator pays more attention on the potentially biased instances thus has smaller influence on the hard instances in the training procedure. These models achieve higher accuracies on the hard subset than the baseline models in both datasets. The ‘reweight’ trick in Sec[3.2.3] greatly reduces...
the performance gap between the easy and hard sets as it dynamically adjusts the debiasing strategies (i.e. the weight of training instances in Eq 5 and 6).

4. Related Work

The bias in the data annotation exists in many tasks, e.g. lexical inference (Levy et al., 2015), visual question answering (Goyal et al., 2017), ROC story cloze (Cai et al., 2017) etc. The NLI models are shown to be sensitive to the compositional features in premises and hypotheses (Nie et al., 2019a), data permutations (Schluter and Varab, 2018) and vulnerable to adversarial examples (Iyyer et al., 2018) and crafted stress test (Geiger et al., 2018; Nakt et al., 2018). (Rüdinger et al., 2017) showed hypothesis in SNLI has the evidence of gender, racial and religious stereotypes, etc. (Sanchez et al., 2018) analysed the behaviour of NLI models and the factors to be more robust. (Feng et al., 2019) discussed how to use partial-input baseline (hypothesis-only classifier in NLI) in future dataset creation. (Clark et al., 2019) uses an ensemble-based method to mitigate known bias. The InferSent model, which served as an important baseline in this paper, are found to achieve superb performance on SNLI by word-level heuristics (Dasgupta et al., 2018). (MacCartney and Manning, 2009) first revealed the difficulties of natural language inference model with bag-of-words models. Different from the artificial patterns we used in this paper, other artifact evidence includes sentence occurrence (Zhang et al., 2019), syntactic heuristics between hypotheses and premises (McCoy et al., 2019) and black-box clues derived from neural models (Gururangan et al., 2018; Poliak et al., 2018; He et al., 2019).

The adversarial debiasing training proposed in this paper is inspired by the success of Generative Adversarial Net-works (GANs) (Goodfellow et al., 2014). Several works on learning encoders which are invariant to certain properties of text and image (Chen et al., 2018; Zhang et al., 2017; Xie et al., 2017) Moyer et al., 2018; Jaiswal et al., 2018) in the adversarial settings.

5. Conclusion

In this study, we show that the hypothesis-only bias in trained NLI models mainly comes from unevenly distributed surface patterns, which could be used to identify hard and easy instances for more convincing re-evaluation on currently overestimated NLI models. The attempts to mitigate the bias are meaningful as such bias not only makes NLI models fragile to adversarial examples. We try to mitigate this bias by removing those artificial patterns in the training sets, with experiments showing that it is a feasible way to alleviate the bias under proper down-sampling methods. We also show that adversarial debiasing with the guidance from the harvested artificial patterns is a feasible way to mitigate the hypothesis-only bias for sentence vector-based NLI models.

6. Bibliographical References

Belinkov, Y., Poliak, A., Shieber, S. M., and Van Durme, B. (2018). Mitigating bias in natural language inference using adversarial learning.

Belinkov, Y., Poliak, A., Shieber, S. M., Van Durme, B., and Rush, A. M. (2019). Don’t take the premise for granted: Mitigating artifacts in natural language inference. arXiv preprint arXiv:1907.04380.

Cai, Z., Tu, L., and Gimpel, K. (2017). Pay attention to the ending: Strong neural baselines for the ROC story cloze task. In ACL.

Chen, Q., Zhu, X., Ling, Z.-H., Inkpen, D., and Wei, S. (2017a). Neural natural language inference models enhanced with external knowledge. arXiv preprint arXiv:1711.04289.

Chen, Q., Zhu, X., Ling, Z., Wei, S., Jiang, H., and Inkpen, D. (2017b). Enhanced LSTM for natural language inference. In ACL.

Chen, X., Sun, Y., Athiwaratkun, B., Cardie, C., and Weinberger, K. (2018). Adversarial deep averaging networks for cross-lingual sentiment classification. TACL, 6:557–570.

Clark, C., Yatskar, M., and Zettlemoyer, L. (2019). Don’t take the easy way out: Ensemble based methods for avoiding known dataset biases. arXiv preprint arXiv:1909.03683.

Conneau, A., Kiela, D., Schwenk, H., Barrault, L., and Bordes, A. (2017). Supervised learning of universal sentence representations from natural language inference data. In EMNLP, pages 670–680.
7. Language Resource References

Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. (2015). A large annotated corpus for learning natural language inference. In EMNLP. Association for Computational Linguistics.

Williams, A., Nangia, N., and Bowman, S. (2018). A broad-coverage challenge corpus for sentence understanding through inference. In NAACL, pages 1112–1122. Association for Computational Linguistics.