Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 6852–6860
Marseille, 11–16 May 2020

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

Tianyu Liu[1], Xin Zheng[3], Baobao Chang[1,2], Zhifang Sui[1,2]
[1]MOE Key Lab of Computational Linguistics, School of EECS, Peking University
[2]Peng Cheng Laboratory, Shenzhen, China [3]Beijing University of Posts and Telecommunications
tianyu0421@pku.edu.cn, zheng_xin@bupt.edu.cn, chbb@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 can not 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 hypothese 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., 2017b) and popular pretrained models like BERT (Devlin et al., 2018), XLNet (Yang et al., 2019) and RoBERTa (Liu et al., 2019), 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 reducing 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

	Multi-word Patterns						Unigram Patterns					
	Entailment		Neutral		Contradiction		Entailment		Neutral		Contradiction	
SNLI	in this picture	96.4	tall human	99.7	Nobody # # .	99.8	outdoors	78.8	vacation	91.0	Nobody	99.7
	A human	96.4	A sad	95.6	dog # sleeping	97.5	sport	75.1	winning	89.9	No	95.8
	<u>A # # outdoors .</u>	95.9	A # human	94.1	There # no	96.2	instrument	74.4	favorite	88.7	cats	93.4
	A # # outside .	89.8	the first	88.6	in # bed	94.2	animal	68.5	date	87.4	naked	88.7
	is near # # .	87.6	on # way	87.0	at home	93.5	moving	67.8	brothers	85.6	tv	88.4
MultiNLI	It # possible	71.7	, said the	93.6	There are no	92.4	Several	54.7	addition	69.6	None	85.4
	There # a # # the	70.8	They wanted to	81.4	does not # any	91.9	Yes	54.4	also	68.6	refused	80.5
	There is an	68.8	the most popular	78.7	no # on	91.5	various	53.7	locals	65.7	never	79.0
	are two	67.0	addition to	78.4	are any	90.1	...	53.1	battle	63.3	perfectly	77.3
	There # some	65.9	because he was	77.8	are never	89.9	According	53.1	dangerous	63.2	Nobody	77.1

Table 1: Top 3 artificial patterns sorted by the pattern-label conditional probability $\mathrm{p}(l|b)$ (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..

al., 2018; Luo et al., 2018). The experiments show that the guidance from the derived artificial patterns can be helpful to the success of sentence-level NLI debiasing.

2. Datasets

In this section, we identify the artificial patterns from the hypothesis sentences which highly correlate to specific labels in the training sets and then derive hard, easy subsets from the original test sets based on them.

2.1. 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 $\mathrm{H}(M, t, \lambda)$ 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 $b \in \mathrm{H}(M, t, \lambda)$, there exists a specific label l for b that the conditional probability $\mathrm{p}(l|b) = count(b, l)/count(b) > \lambda$. For example, for the underlined pattern 'A # # outdoors .' in Table 1, the length of this pattern is 3, and the distance between the consecutive words 'A' and 'outdoors' 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[1].

In the rest of paper, unless otherwise specified, we set $M = 3, t = 3$ [2]. By doing so, we only tune the hyper-parameter λ in H(3,3,λ) to decide using a rather strict (smaller λ) or mild (bigger λ) strategy while deriving the artificial patterns.

2.2. 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, H(3,3,0.5)) 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.

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

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..

2.3. 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

[1] Suppose a pattern only appears once in a training instance, its $\mathrm{p}(l|b)$ always equals 1 for the label in that instance.

[2] We also tried larger M and t, e.g. 4 or 5, but did not observe considerable changes of the artificial patterns, e.g. 95.4% patterns in H(5,5,0.5) are covered by H(3,3,0.5).

Premise: Two cats playing on the bed together .
Hypothesis: *The dogs are* playing *on* the bed .
Gold Label: CONTRADICTION
Artificial patterns: (bed ., CONTRADICTION, 83.2%); (*The dogs are # on*, CONTRADICTION, 82.9%)

(a) An easy instance

Premise: A bare-chested man fitting his head and arm into a toilet seat ring while spectators watch in a city.
Hypothesis: A gentleman with no chest hair , wrangles *his way* through a toilet seat .
Gold Label: ENTAILMENT
Artificial patterns: (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 $(\mathbf{P}, l, p)$ show the related label indications l for specific artificial patterns $\mathbf{P}$ and their conditional probability p.

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'[3] 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. λ in H$(M = 3, t = 3, \lambda)$ (Sec 2.1. and Footnote 2). For the sake of simplicity, we utilize $\lambda = 0.8$ and $\lambda = 0.7$ [4] as the thresholds to derive easy and hard subsets for SNLI and MultiNLI respectively in the following experiments, as adopting a relatively bigger λ 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. [5] 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.

	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

	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

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 detected by a neural based hypothesis-only classifier(Gururangan et al., 2018).

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 hyperparameter and optimizer settings, including InferSent[6], DAM[7], ESIM[8], BERT (uncased base), XLNet (cased base) 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 (Gururangan et al., 2018) which are derived from a hypothesis-only classifier. From 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] '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 λ 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

λ	No.	Mode	Full	Easy	Hard	$\Delta_{Easy}^{Hard}(\downarrow)$
0.5	I-1	Rand	**76.4**	**93.8**	48.7	45.1
	I-2	Debias	66.9	64.6	**56.0**	8.6
0.6	I-3	Rand	**81.1**	**96.1**	54.1	42.0
	I-4	Debias	76.9	79.8	**58.0**	21.8
0.7	I-5	Rand	**82.8**	**96.9**	56.0	40.9
	I-6	Debias	80.9	86.4	**61.5**	24.9
0.8	I-7	Rand	**83.5**	**96.9**	56.6	40.3
	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) InferSent trained on SNLI

λ	No.	Mode	Full	Easy	Hard	$\Delta_{Easy}^{Hard}(\downarrow)$
0.5	D-1	Rand	**74.1**	**92.3**	46.7	45.6
	D-2	Debias	67.5	67.8	**53.3**	14.5
0.6	D-3	Rand	**82.4**	**96.3**	56.7	39.6
	D-4	Debias	79.3	84.4	**62.1**	21.3
0.7	D-5	Rand	**84.4**	**97.3**	59.4	37.9
	D-6	Debias	83.0	89.6	**63.1**	26.5
0.8	D-7	Rand	**85.3**	**97.8**	60.1	37.7
	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

λ	No.	Mode	Full	Easy	Hard	$\Delta_{Easy}^{Hard}(\downarrow)$
0.5	E-1	Rand	**76.8**	**94.6**	48.9	45.7
	E-2	Debias	65.3	62.5	**53.8**	7.7
0.6	E-3	Rand	**83.6**	**96.6**	62.2	34.4
	E-4	Debias	78.6	79.4	**63.6**	15.8
0.7	E-5	Rand	**85.9**	**97.2**	64.2	35.0
	E-6	Debias	83.8	88.2	**68.8**	19.4
0.8	E-7	Rand	**86.9**	**97.3**	67.9	29.4
	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

Table 5: Model performance on the SNLI test set. We report the average scores of multiple independent runs. Δ_{Easy}^{Hard} represents the gap between hard and easy test sets (lower is better). λ 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 λ=1.0, we use the whole training set without any downsampling.

λ	No.	Mode	Full	Easy	Hard	$\Delta_{Easy}^{Hard}(\downarrow)$
0.5	I-1	Rand	**67.5**	**92.6**	50.9	41.7
	I-2	Debias	64.2	76.0	**56.1**	19.9
0.6	I-3	Rand	**69.0**	**92.7**	53.4	39.3
	I-4	Debias	67.5	80.9	**59.7**	21.2
0.7	I-5	Rand	**69.1**	**93.0**	52.2	40.8
	I-6	Debias	68.3	84.6	**57.1**	27.5
0.8	I-7	Rand	69.2	**92.5**	52.5	40.0
	I-8	Debias	**69.6**	91.4	**53.6**	37.8
1.0	I-9	All	70.4	92.7	54.4	38.3

(a) InferSent trained on MultiNLI

λ	No.	Mode	Full	Easy	Hard	$\Delta_{Easy}^{Hard}(\downarrow)$
0.5	D-1	Rand	**64.4**	**91.9**	46.2	45.7
	D-2	Debias	61.9	77.4	**52.0**	22.4
0.6	D-3	Rand	**68.3**	**91.8**	52.5	39.3
	D-4	Debias	65.8	79.0	**58.0**	21.0
0.7	D-5	Rand	**69.6**	**92.6**	52.2	40.4
	D-6	Debias	68.0	83.9	**57.5**	25.4
0.8	D-7	Rand	**70.0**	**92.4**	53.8	38.6
	D-8	Debias	69.6	91.6	**54.9**	26.7
1.0	D-9	All	70.5	92.0	55.1	36.9

(b) DAM trained on MultiNLI

λ	No.	Mode	Full	Easy	Hard	$\Delta_{Easy}^{Hard}(\downarrow)$
0.5	E-1	Rand	**70.4**	**92.8**	59.0	33.8
	E-2	Debias	67.0	75.2	**63.6**	11.6
0.6	E-3	Rand	**73.9**	**93.8**	61.6	22.2
	E-4	Debias	73.2	84.8	**66.4**	18.4
0.7	E-5	Rand	74.6	**92.8**	64.5	28.3
	E-6	Debias	**74.9**	89.1	**65.9**	23.2
0.8	E-7	Rand	75.7	**94.0**	64.4	29.6
	E-8	Debias	**75.8**	93.4	**65.0**	28.4
1.0	E-9	All	76.7	93.9	65.6	28.3

(c) ESIM trained on MultiNLI

Table 6: Models performance on MultiNLI mismatched dev set. We tune the models on MultiNLI matched dev set. The training sizes are 24.4%, 53.1%, 68.0%, 81.2% 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 λ. 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 λ: The threshold λ is exactly the same λ 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 λ, 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 λ represents more strict down-sampling strategy in terms of filtering the artificial patterns. $\lambda = 0.5$ serves as the lower bound because the highest pattern-label conditional probability (p($l|b$) in Sec 2.1.) for premises, which aren't observed the same bias as hypotheses, is less than 0.5 in both SNLI and MultiNLI training set..

Ruling out the effects of training size: The model performance might be highly correlated with the size of training set. To rule out the effects of training size as much as possible, we set up randomly down-sampled training sets ('rand' mode) with the same size as the corresponding 'debias' mode under different λ for a fair comparison.

Keeping the label distribution balanced: After removing the biased instances in the training set by different λ ('debias' mode), suppose we get n_1, n_2, n_3 ($n_1 \geq n_2 \geq n_3$) instances for the 3 pre-defined labels of NLI in the down-sampled training set. Then we down-sample the subsets with n_1, n_2 instances to n_3 instances and get a dataset with $3n_3$ instances. For the corresponding 'rand' mode, we sample 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 λ for 5 times and get 5 randomly down-sampled training sets for each λ. 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 λ, 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 really impressive as the training size of I-8, I-8 and E-8 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 $\mathrm{p}(l|b)$ 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.

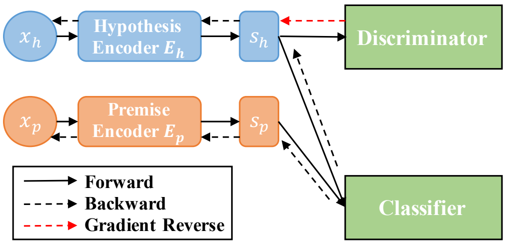

Figure 1: The illustration of the sentence-level debiasing framework, which is elaborated in Sec 3.2.1..

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' [10] category as defined on SNLI's web page[11]. 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 $s_h = \mathrm{E}_h(x_h), s_p = \mathrm{E}_s(x_s)$ of hypothesis and premise encoders $\mathrm{E}_h, \mathrm{E}_p$, we are interested in predicting the NLI label y using a classifier C, $\mathrm{p_C}(y|s_h = \mathrm{E}_h(x_h), s_p = \mathrm{E}_s(x_s))$. 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 $\mathrm{p_D}(y|s_h = \mathrm{E}_h(x_h))$. 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. γ is a hyper-parameter which controls the degree of debiasing.

$$\begin{aligned}\min_{\mathrm{E}_h,\mathrm{E}_p,\mathrm{C}} \max_{\mathrm{D}} E_{x_h,x_p,y\sim\mathrm{p}(X,Y)}[\gamma \log \mathrm{p_D}(y|\mathrm{E}_h(x_h)) - \\ \log \mathrm{p_C}(y|\mathrm{E}_h(x_h), \mathrm{E}_p(x_p))]\end{aligned} \tag{1}$$

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 (Xie et al., 2017;

[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 θ by gradient descending (m is the batch size):

$$\theta_{\mathrm{D}}^{new} = \theta_{\mathrm{D}}^{old} - \frac{1}{m}\sum_{i=1}^{m}\nabla_{\theta_{\mathrm{D}}}[\log \mathrm{p}_{\mathrm{D}}(y^i|\mathrm{E}_h(x_h^i))] \tag{2}$$

$$\theta_{\mathrm{C}}^{new} = \theta_{\mathrm{C}}^{old} - \frac{1}{m}\sum_{i=1}^{m}\nabla_{\theta_{\mathrm{C}}}[\log \mathrm{p}_{\mathrm{C}}(y^i|\mathrm{E}_h(x_h^i), \mathrm{E}_p(x_p^i))] \tag{3}$$

$$\begin{aligned}\theta_{\mathrm{E}_h}^{new} = \theta_{\mathrm{E}_h}^{old} - \frac{1}{m}\sum_{i=1}^{m}\nabla_{\theta_{\mathrm{E}_h}}[\log \mathrm{p}_{\mathrm{C}}(y^i|\mathrm{E}_h(x_h^i), \mathrm{E}_p(x_p^i))]\\ \underbrace{+\frac{\gamma}{m}\sum_{i=1}^{m}\nabla_{\theta_{\mathrm{E}_h}}[\log \mathrm{p}_{\mathrm{D}}(y^i|\mathrm{E}_h(x_h^i))]}_{\textbf{gradient reverse}}\end{aligned} \tag{4}$$

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:

$$\begin{aligned}\frac{1}{m}\sum_{i=1}^{m}[1\{(x^i, y^i) \in \mathcal{D}_{unbias}\}\log \mathrm{p}_{\mathrm{D}}(y^i|\mathrm{E}_h(x_h^i)) +\\ \alpha_1 * 1\{(x^i, y^i) \in \mathcal{D}_{bias}\}\log \mathrm{p}_{\mathrm{D}}(y^i|\mathrm{E}_h(x_h^i))]\end{aligned} \tag{5}$$

, where $\mathcal{D} = \mathcal{D}_{unbias} \cup \mathcal{D}_{bias}$ denotes the whole training corpus. The division of biased and unbiased training subsets depends on the debiasing threshold λ (just like the down-sampling threhold in Table 5 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.

$$\begin{aligned}\frac{1}{m}\sum_{i=1}^{m}[1\{(x^i, y^i) \in \mathcal{D}_{bias}\}\log \mathrm{p}_{\mathrm{C}}(y^i|\mathrm{E}_h(x_h^i), \mathrm{E}_h(x_p^i)) +\\ \alpha_2 * 1\{(x^i, y^i) \in \mathcal{D}_{unbias}\}\log \mathrm{p}_{\mathrm{C}}(y^i|\mathrm{E}_h(x_h^i), \mathrm{E}_h(x_p^i))]\end{aligned} \tag{6}$$

, 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 α (α_1 and α_2 in Eq 5 and Eq 6 respectively), we find it useful to adjust α dynamically in the training process. α^0 and α^t represents the initial values we set before training and its value after t training iterations respectively. Additionally we set up a hyper-parameter ϕ to control the gap of model accuracies δ on the easy and hard subsets in the dev set.

$$\alpha^{t+1} = \begin{cases} \max(\alpha^t + \epsilon, \alpha^0), & \delta \geq \phi \\ \max(\alpha^t - \epsilon, \alpha^0), & \delta < \phi \end{cases} \tag{7}$$

where ϵ is a hyper-parameter set as 0.5 for models trained on both datasets. Besides, we set ϕ as 0.15 and 0.10 for SNLI and MultiNLI respectively. Notably although we update the hyper-parameters α_1 and α_2 dynamically in different iterations based on ϕ, 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, 6 and Eq 1. We also try $\lambda \in \{0.5, 0.6, 0.7, 0.8\}$ as the threshold to split $\mathcal{D}_{bias}$ and $\mathcal{D}_{unbias}$ in Eq 5 and 6. Specifically, we treat the instances which contain the artificial patterns in H(3,3,λ) (Sec 2.1., Footnote 2) as $\mathcal{D}_{bias}$, and the remaining instances as $\mathcal{D}_{unbias}$. For the results in Table 7, we set $\gamma = 3$ and $\gamma = 1$ for SNLI and MultiNLI respectively. For both datasets, we set $\alpha_1 = 5, \alpha_2 = 5$ as well as $\lambda = 0.7$ 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

Model	Full	Easy	Hard	$\Delta^{Hard}_{Easy}(\downarrow)$
InferSent	**84.5**	**97.2**	58.9	38.3
InferSent+DS(λ=0.8)	82.9	90.4	60.0	30.4
InferSent+Guidance	84.1	95.5	**61.7**	33.8
dInferSent	81.6	**92.5**	59.9	32.6
+Guidance	**82.2**	86.9	63.3	23.6
+Guidance+Reweight	80.9	78.2	**67.3**	10.9

(a) InferSent trained on SNLI

Model	Full	Easy	Hard	$\Delta^{Hard}_{Easy}(\downarrow)$
InferSent	**70.4**	**92.7**	54.4	38.3
InferSent+DS(λ=0.8)	69.9	91.4	53.6	37.8
InferSent+Guidance	70.1	92.1	**54.9**	37.2
dInferSent	**68.8**	**91.1**	54.7	36.4
+Guidance	68.0	87.9	55.3	32.6
+Guidance+Reweight	66.5	79.4	**58.8**	20.6

(b) InferSent trained on MultiNLI

Table 7: The comparison of InferSent (baseline), InferSent+DS (downsampling) and dInferSent (adversarial debiasing) on SNLI test set and MultiNLI mismatched dev set respectively. We choose the down-sampling (DS) method with $\lambda = 0.8$ because it performs best on the hard subsets. The 'Guidance' and 'Reweight' methods are elaborated in Sec 3.2.2. and Sec 3.2.3. respectively.

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; Wang et al., 2018) and vulnerable to adversarial examples (Iyyer et al., 2018; Minervini and Riedel, 2018; Glockner et al., 2018) and crafted stress test (Geiger et al., 2018; Naik et al., 2018). (Rudinger 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 Networks (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.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable suggestions. This work is supported by the National Science Foundation of China under Grant No. 61751201, No. 61772040, No. 61876004. The corresponding authors of this paper are Baobao Chang and Zhifang Sui.

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.

Dagan, I., Glickman, O., and Magnini, B. (2006). The pascal recognising textual entailment challenge. pages 177–190. Springer.

Dagan, I., Roth, D., Sammons, M., and Zanzotto, F. M. (2013). *Recognizing Textual Entailment: Models and Applications*. Synthesis Lectures on Human Language Technologies.

Dasgupta, I., Guo, D., Stuhlmüller, A., Gershman, S. J., and Goodman, N. D. (2018). Evaluating compositionality in sentence embeddings. *arXiv preprint arXiv:1802.04302*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Feng, S., Wallace, E., and Boyd-Graber, J. (2019). Misleading failures of partial-input baselines. *arXiv preprint arXiv:1905.05778*.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. (2016). Domain-adversarial training of neural networks. *JMLR*, 17(1):2096–2030.

Geiger, A., Cases, I., Karttunen, L., and Potts, C. (2018). Stress-testing neural models of natural language inference with multiply-quantified sentences. *arXiv preprint arXiv:1810.13033*.

Glockner, M., Shwartz, V., and Goldberg, Y. (2018). Breaking NLI systems with sentences that require simple lexical inferences. In *ACL*, pages 650–655.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *NIPS*, pages 2672–2680.

Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., and Parikh, D. (2017). Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In *CVPR*.

Gururangan, S., Swayamdipta, S., Levy, O., Schwartz, R., Bowman, S. R., and Smith, N. A. (2018). Annotation artifacts in natural language inference data. In *NAACL*, pages 107–112.

He, H., Zha, S., and Wang, H. (2019). Unlearn dataset bias in natural language inference by fitting the residual. *arXiv preprint arXiv:1908.10763*.

Iyyer, M., Wieting, J., Gimpel, K., and Zettlemoyer, L. (2018). Adversarial example generation with syntactically controlled paraphrase networks. In *NAACL*, pages 1875–1885.

Jaiswal, A., Wu, R. Y., Abd-Almageed, W., and Natarajan, P. (2018). Unsupervised adversarial invariance. In *NIPS*, pages 5097–5107.

Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., and Mikolov, T. (2016). Fasttext. zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Levy, O., Remus, S., Biemann, C., and Dagan, I. (2015). Do supervised distributional methods really learn lexical inference relations? In *NAACL*, pages 970–976.

Lin, Z., Feng, M., Santos, C. N. d., Yu, M., Xiang, B., Zhou, B., and Bengio, Y. (2017). A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Luo, F., Liu, T., He, Z., Xia, Q., Sui, Z., and Chang, B. (2018). Leveraging gloss knowledge in neural word sense disambiguation by hierarchical co-attention. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1402–1411.

MacCartney, B. and Manning, C. D. (2009). *Natural language inference*. Citeseer.

McCoy, R. T., Pavlick, E., and Linzen, T. (2019). Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. *arXiv preprint arXiv:1902.01007*.

Minervini, P. and Riedel, S. (2018). Adversarially regularising neural NLI models to integrate logical background knowledge. In *CoNLL*, pages 65–74.

Moyer, D., Gao, S., Brekelmans, R., Steeg, G. V., and Galstyan, A. (2018). Evading the adversary in invariant representation. *arXiv preprint arXiv:1805.09458*.

Naik, A., Ravichander, A., Sadeh, N., Rosé, C. P., and Neubig, G. (2018). Stress test evaluation for natural language inference. In *COLING*, pages 2340–2353.

Nie, Y., Wang, Y., and Bansal, M. (2019a). Analyzing compositionality-sensitivity of NLI models. *AAAI*.

Nie, Y., Williams, A., Dinan, E., Bansal, M., Weston, J., and Kiela, D. (2019b). Adversarial nli: A new benchmark for natural language understanding. *arXiv preprint arXiv:1910.14599*.

Parikh, A. P., Täckström, O., Das, D., and Uszkoreit, J. (2016). A decomposable attention model for natural language inference. In *EMNLP*, pages 2249–2255.

Poliak, A., Naradowsky, J., Haldar, A., Rudinger, R., and Durme, B. V. (2018). Hypothesis only baselines in natural language inference. In **SEM@NAACL-HLT*, pages 180–191.

Rudinger, R., May, C., and Durme, B. V. (2017). Social bias in elicited natural language inferences. In *EthNLP@EACL*, pages 74–79.

Sanchez, I., Mitchell, J., and Riedel, S. (2018). Behavior analysis of nli models: Uncovering the influence of three factors on robustness. In *EMNLP*, volume 1, pages 1975–1985.

Schluter, N. and Varab, D. (2018). When data permutations are pathological: the case of neural natural language inference. In *EMNLP*, pages 4935–4939.

Tsuchiya, M. (2018). Performance impact caused by hidden bias of training data for recognizing textual entailment. In *LREC*.

Wang, H., Sun, D., and Xing, E. P. (2018). What if we simply swap the two text fragments? a straightforward yet effective way to test the robustness of methods to confounding signals in nature language inference tasks. *arXiv preprint arXiv:1809.02719*.

Wu, W., Wang, H., Liu, T., and Ma, S. (2018). Phrase-level self-attention networks for universal sentence encoding. In *Proceedings of the 2018 Conference on Em-*

pirical Methods in Natural Language Processing, pages 3729–3738.

Xie, Q., Dai, Z., Du, Y., Hovy, E., and Neubig, G. (2017). Controllable invariance through adversarial feature learning. In *NIPS*.

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., and Hovy, E. (2016). Hierarchical attention networks for document classification. In *NAACL 2016*, pages 1480–1489.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., and Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

Zellers, R., Bisk, Y., Schwartz, R., and Choi, Y. (2018). Swag: A large-scale adversarial dataset for grounded commonsense inference. *arXiv preprint arXiv:1808.05326*.

Zhang, Y., Barzilay, R., and Jaakkola, T. (2017). Aspect-augmented adversarial networks for domain adaptation. *TACL*, 5:515–528.

Zhang, G., Bai, B., Liang, J., Bai, K., Chang, S., Yu, M., Zhu, C., and Zhao, T. (2019). Selection bias explorations and debias methods for natural language sentence matching datasets. *arXiv preprint arXiv:1905.06221*.

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.