Improving the Natural Language Inference Robustness to Hard Dataset by Data Augmentation and Preprocessing

Zijiang YANG[1]

[1] University of Texas at Austin, 2317 Speedway, GDC 2.302 Austin, Texas 78712, USA
zy4957@eid.utexas.edu

Abstract. Natural Language Inference (NLI) is the task of inferring whether the hypothesis can be justified by the given premise. Basically, we classify the hypothesis into three labels (entailment, neutrality and contradiction) given the premise. NLI was well studied by the previous researchers. A number of models, especially the transformer-based ones, have achieved significant improvement on these tasks. However, it is reported that these models are suffering when they are dealing with hard datasets. Particularly, they perform much worse when dealing with unseen out-of-distribution premise and hypothesis. They may not understand the semantic content but learn the spurious correlations. In this work, we propose the data augmentation and preprocessing methods to solve the word overlap, numerical reasoning and length mismatch problems. These methods are general methods that do not rely on the distribution of the testing data and they help improve the robustness of the models. The experimental results show that the proposed method achieves more than 12% classification accuracy improvement on the HANS dataset and 6% to 9% improvement on the ANLI dataset. The notable improvement is achieved by only training on extra 3% data.

Keywords: Natural Language Inference, Robustness, Data Augmentation, Preprocessing

1 Introduction

The Natural Language Inference (NLI) is the task of recognizing the textual entailment by determining the relationship between the hypothesis and the given premise.

It is a classification task that returns the label *entailment(0)/contradiction(2)* when the hypothesis agrees/disagrees with the premise. If the correctness of the hypothesis is unknown given the context of the premise, this task will return the label *neutrality(1).* The NLI task has been developed as a benchmark task for natural language understanding research [1],[2]. The objective of this task is to let the natural language models understand the contextual semantics and make logical decisions.

A lot of models have been proposed to solve this NLI task: BERT [3], RoBERTa [4] XLNet [5]. The most recent state-of-the-art model is the Masked Language

Modeling(MLM) pre-training models like BERT or its variants. These models understand the contextual relationships by masking out the token and reconstructing it back. It is shown to be effective in the downstream task like the NLI task. Recently the pre-trained model ELECTRA-small [6] gained much popularity due to its computational efficiency in solving the downstream task.

These transformer-based models consistently achieve high accuracies on the simple datasets. However, such models may not really do the inference tasks. Rather, they may simply perform the 'pattern matching' between the testing data and the training data [7], [8]. One simple example would be the word overlap feature. The model may just compute the statistics of word overlap between the premise and hypothesis. The classification result will be entailment if the overlapping is significant. Such process is definitely not inference but memorization. To stress test and evaluate the capability of the models, more sophisticated datasets were generated by previous researchers. They deliberately generated data that tend to fool the model but not the human readers. The model performance on these hard datasets is not satisfactory with accuracies below 50%.

To address this problem, we analyze the limitations of current model on three aspects: (1) word overlap; (2) numerical reasoning; (3) length mismatch. Several methods are proposed to tackle these problems.

The paper selects the ELECTRA-small as the base model for conducting the analysis. We will focus on its performance on the simple dataset SNLI and then evaluate the model on other hard datasets. The paper is structured as follows: Section 1 introduces the background information; Section 2 discusses the weak robustness of the model. Section 3 will analyze the associated problems and propose several improvement methods. The experimental results will be fully analyzed. Finally, we make the conclusion for the paper in Section 4.

2 Weak Robustness of the NLI models

Previous researcher have done a lot of work on the error analysis of the best-performing transformer model on different datasets [9],[10]. They have discovered the weak robustness of the models. To analyze the robustness of the models, we trained the model(ELECTRA-small) on the benchmark dataset, SNLI, whose information can be found in the section 3.1. We discovered that the model cannot actually conduct the real inference even on the simplest form of premise-hypothesis pairs, if such pairs are in the out-of-distribution. Based on our analysis of the error types, we can conclude the followings are the main common mistake the model would make:

- **Word overlap**: Large overlap between the premise and the hypothesis causes wrong classification decision. For the example on the text-box, we can that the premise *'The student supports the teacher.'* and the hypothesis *'The teacher supports the student.'* have 100% word overlap. However, the model will predict the label entailment with

probability close to 1. Obviously, in this example, the model cannot learn the relationship between the subject and the object.

- **Numerical reasoning**: Another difficulty with the model is its numerical reasoning capability. When it comes to reasoning about the chronological orders of the events, it often gives very strange answers. Like the example in the text-box, it gives completely opposite label, contradiction for the gold label vs. entailment for the model result. Also, we find that the model cannot perform well on the numbers counting, comparison and basic arithmetic tasks.
- **Length mismatched**: The premise may be much longer than the hypothesis. The information may be hidden inside only one sentence in the premise, while the other sentences can be the distraction for the model. Take the example in the text-box. The whole paragraph is too long for the model to understand the location of *'The Winds of Whoopie'*. The model gets puzzled and produces the label neutrality. However, if we remove the first sentence of the premise, the model result will be entailment. Moreover, if we extract the important information out from the premise and generate a new premise: *'He produced the television special 'The Winds of Whoopie''*. The probability for predicting entailment will be much higher, which is close to 1.

Misclassification of word overlap example

P: The student supports the teacher.
H: The teacher supports the student.

Gold label: Neutrality(1)
Model result: Entailment(0)

Misclassification of numerical reasoning example

P: He (born in 1935) was an actor.
H: He was born before 1934.

Gold label: Contradiction(2)
Model result: Entailment(0)

Misclassification of length mismatched example

P: Alan Metter is an American film director whose most notable credits include "Back to School" starring Rodney Dangerfield, and "Girls Just Want to Have Fun" with Sarah Jessica Parker. He also produced and directed the 1983 television special "The Winds of Whoopie" for Steve Martin.

H: "The Winds of Whoopie" was not produced to play in theaters.
Gold label: Entailment(0) **Model result: Neutrality(1)**

2.1 Discussion

Based on the out-of-distribution examples, we can observe that the model exhibits weak robustness. The NLI models are prone to learning the superficial relationship between the premise and the hypothesis such as the statistics of the world overlapping ratio. The model tends to memorize but not infer the relationships. In the following sections, we would train the model to improve its robustness via data augmentation and data preprocessing.

3 Improve the NLI Models' Robustness

3.1 Experimental Data

We include various sets of data to test how the models can learn from the data and how well they can perform on the in-distribution and out-of-distribution data. The data used in this experiment include the following datasets. The difficulty is in the increasing order:

- SNLI [2]: Stanford Natural Language Inference dataset is a widely used dataset for training and evaluating the NLI model. It contains a collection of 570,000 human-written English sentence pairs for balanced classification.
- MultiNLI [11]: Multi-Genre Natural Language Inference dataset is a crowd-sourced collection of 433,000 sentence pairs with annotation. It is an extension and more diversion version of the SNLI. While SNLI mainly focuses on the image captions, the MultiNLI captures a variety of genres including fiction, letters, speech, report and so on.
- HANS [12]: It is mainly for evaluating the NLI models on the specific heuristic but invalid hypotheses. These heuristics are likely to be learned by the model but there are not the real inference.
- ANLI [13]: It is designed to challenge the NLI models by introducing adversarial examples to the dataset. The data is collected via an iterative, adversarial human-model procedure. Its main goal is to try to fool the model with premise-hypothesis sentence pairs. It contains 3 rounds of datasets(r1, r2 and r3). r3 is built on top of r2 and r2 is built on top of r1. In terms of difficulty: $r3 > r2 > r1$

3.2 Methodology

To improve the robustness of the NLI model on the above-mentioned three aspects. Different methods are proposed:

Word Overlap. We train the models via augmenting simple premise-hypothesis pairs that have large overlapping. We construct the additional data via the following steps:

- Extract out the words in the training corpus that have part-of-speech tag *Noun(NN)* and name it by set *N*.
- Extract out the words in the training corpus that have part-of-speech *Verb(V)* and name it by set *V*.
- Randomly pick two words n_1, n_2 from *N* and one word v_1 from *V*.
- Construct the additional dataset by generating 3 premises: (1) *The* n_1 v_1 *the* n_2. (2) *The* n_1 *does not* v_1 *the* n_2. (3) *The* n_2 v_1 *the* n_1. And the hypothesis will be the *The* n_1 v_1 *the* n_2. So we basically construct 3 pairs with *entailment/contradiction/neutrality* labels.

Numerical Reasoning. Improving numerical reasoning is a difficult task that includes many aspects. We will focus on improving the reasoning about the chronological order of the events. We solve it by augmenting simple premise-hypothesis pairs that ask for comparing the years. The pairs are generated by the following steps:

- Randomly generate an integer i_1 from 0 to 2020 for birth year.
- Randomly generate an integer i_2 from i_1+1 to i_1+100 for death year.
- The premise will be: *He (*i_1 - i_2*)*.
- Randomly generate an integer $i3$ from i_1-100 to i_1+100.
- The hypothesis will be: *He was born before* i_3. The label will be entailment if $i_3>i_1$ else contradiction.
- Similar hypothesis can be : *He was born after* i_4. *He died before* i_5. *He died after* i_6. The labels are assigned accordingly.

Length Mismatch. Length mismatch problem is a built-in problem for the transformer type of models. The length of the input sentence is kept fixed before entering the model. Short sentences will be padded while the long sentences will be truncated. The situation is much worse for the long sentences as some useful and important information will be discarded during the training and evaluation process. Even if we adjust the designated input length for the model, it still cannot handle the long paragraphs. We advocate that for the paragraph input, it is advised to split the paragraph into sentences and then test the hypothesis for each sentence. If any sentence was found to be strongly agreeing or disagreeing the hypotheses, we return the entailment/contradiction label. Otherwise, we return the neutrality label. The details of the *Split* algorithm is shown in Algorithm.1:

Algorithm 1: Split algorithm for paragraph premise

Require: Premise is a paragraph that contains multiple sentences.
Procedure classifyNLI(Hypothesis, Premise)

```
h ← Hypothesis
pList ← strSplit(Premise, '.')
c ← 0.8 // Set the cutoff probability to be 0.8
n ← len(pList)
i ← 0
While i < n do
    V ← Softmax(Model(pList[i], h))
    If V[0] > c then
        Return 0 // Return Entailment
    Else if V[2] > c then
        Return 2 // Return Contradiction
    End If
i ← i + 1
End While
Return 1 // No strong Entailment or Contradiction sentence found, return Neutrality
```

3.3 Experimental Result and Analysis

. We first conduct the experiments on the original SNLI dataset. It is reported that the ELECTRA-small model trained on the SNLI training dataset achieves very high accuracy 0.882 in the SNLI testing dataset, as is shown in the Table.1. However, the model result is bad on the HANS dataset. The model seems to learn the heuristics but not the inference. Additionally, the model performs very poorly on hard dataset ANLI. The accuracy is as low as 0.3, which means the model is almost close to random guessing.

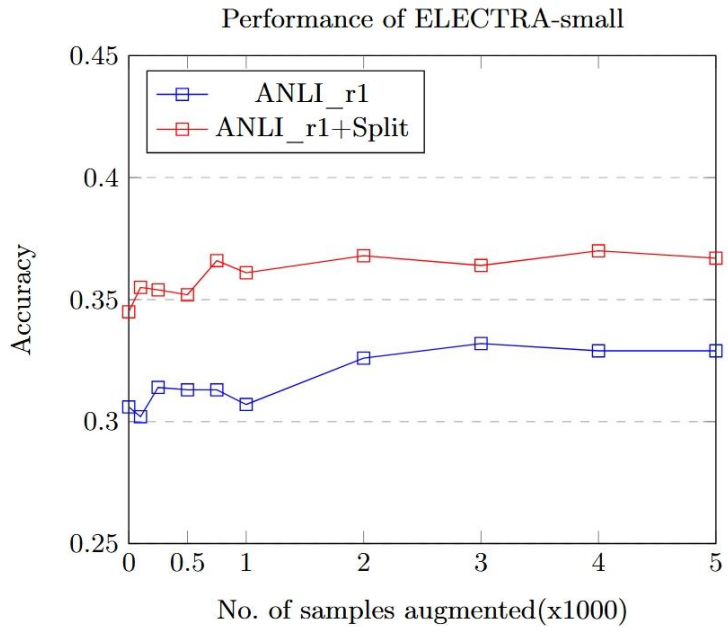
Table 1. Label classification accuracy for models trained on different training data. They are tested on various datasets with different levels of difficulty.

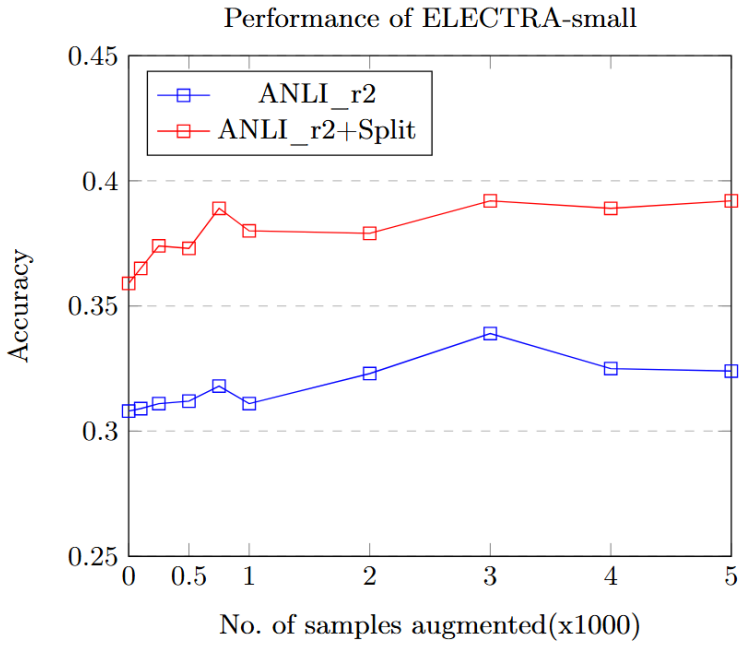
Trained on	*Preprocess technique*	*ANLI*			*HANS*	*MultiNLI*	*SNLI*	*Average*
		r1	r2	r3				
Baseline (simple dataset)								
SNLI	-	0.306	0.308	0.300	0.491	0.706	0.882	0.499
SNLI	+Split	0.345	0.359	0.369	0.494	0.623	0.833	0.502
Proposed								
SNLI(DataAug)	-	0.332	0.339	0.311	0.637	0.660	0.853	0.522
SNLI(DataAug)	+Split	0.359	0.392	0.371	0.616	0.581	0.851	0.528
Baseline (hard dataset)								
ANLI_r1	-	0.373	0.357	0.358	0.492	0.425	0.465	0.412
ANLI_r1	+Split	0.396	0.377	0.359	0.486	0.437	0.443	0.416
ANLI_r2	-	0.460	0.404	0.360	0.489	0.414	0.377	0.417
ANLI_r2	+Split	0.434	0.403	0.358	0.479	0.383	0.362	0.403
ANLI_r3	-	0.471	0.392	0.406	0.398	0.615	0.581	0.477
ANLI_r3	+Split	0.473	0.422	0.395	0.376	0.600	0.591	0.476
Baseline (simple + hard)								
SNLI+ANLI_r1	-	0.413	0.336	0.342	0.488	0.686	0.861	0.521
SNLI+ANLI_r1	+Split	0.374	0.380	0.356	0.498	0.636	0.846	0.515
SNLI+ANLI_r2	-	0.480	0.387	0.360	0.490	0.680	0.831	0.538
SNLI+ANLI_r2	+Split	0.416	0.381	0.376	0.493	0.674	0.843	0.531
SNLI+ANLI_r3	-	0.460	0.396	0.421	0.490	0.724	0.830	0.553
SNLI+ANLI_r3	+Split	0.432	0.395	0.403	0.499	0.689	0.814	0.539

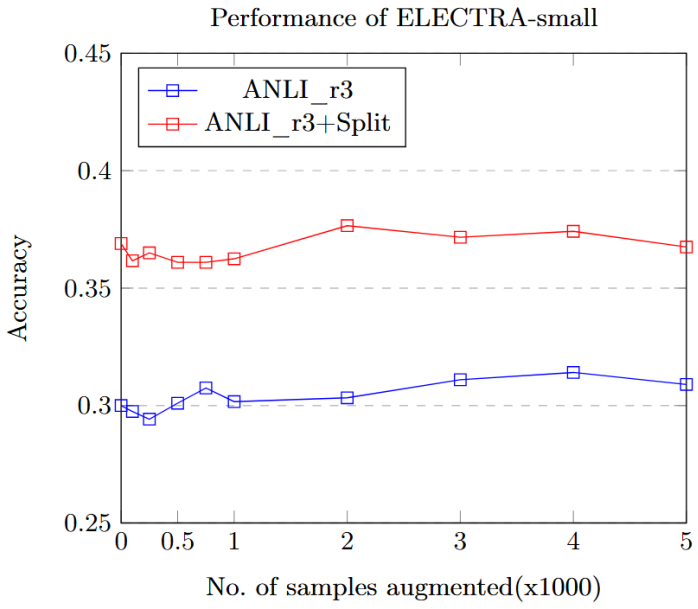
By applying our proposed methods, we first observe a significant accuracy increase on the HANS dataset as we construct the counter-heuristic examples to train the model. The increase is around 14% for data augmentation and 12% for the data augmentation with *Split*. It is worth noting that the performance for the SNLI dataset drops by just 3% by learning some out-of-distribution data.

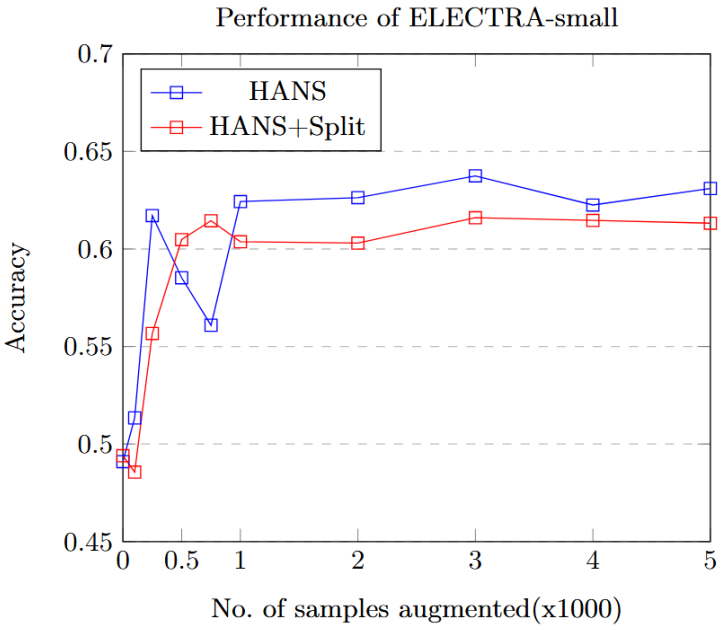
For the model results on the hard datasets ANLI, we see the improvement is substantial. With the data augmentation, the average improvement on ANLI r1,r2,r3 dataset is 3%. With the Split algorithm, the average improvement reaches 5%. By applying both techniques, the average improvement reaches 7%. We see the potential of the model on the hard dataset with the proposed methods. Even though the model is only trained on the simple dataset, the model performance is comparable to the one trained on the hard datasets. The generalization capability and robustness of the model is largely improved.

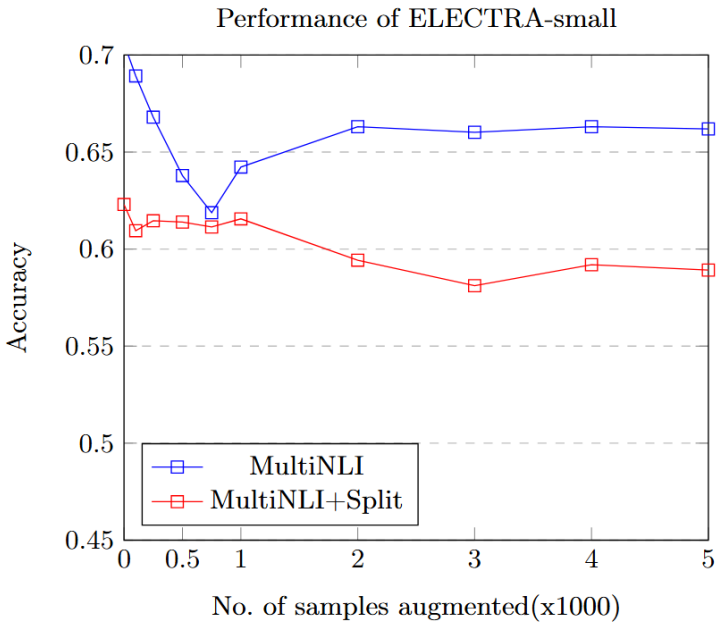
Furthermore, we test our methods by varying the number of augmented examples to the original datasets. The results are shown in the figures below. The first observation will be the obvious accuracy increase on the hard dataset ANLI regardless of the number of augmented samples. The second observation will be the substantial accuracy improvement on HANS by augmenting more samples. Lastly, we observe that the performance improvement reaches the limit at the 1000 augmented samples. Therefore, with only 1000 additional sample, as is compared to 40,000 original training data size, it is sufficient for the proposed method to perform well.

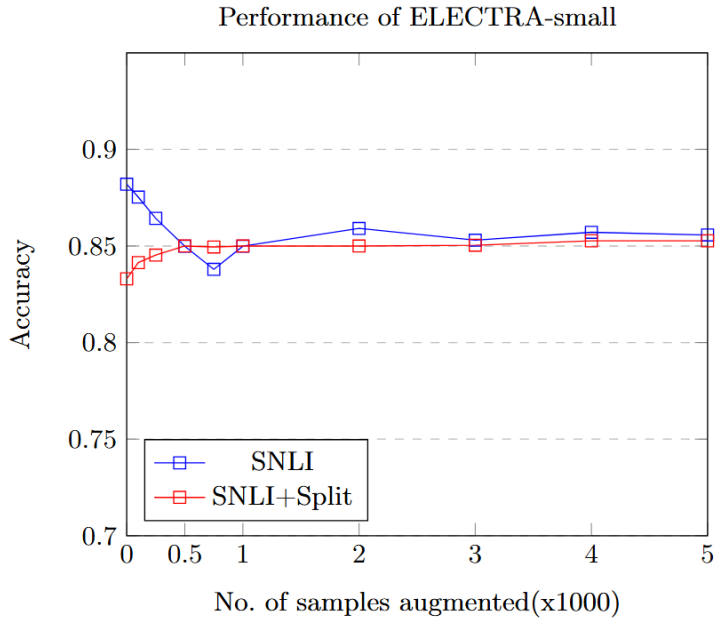

Performance of ELECTRA-small
ANLI_r1
ANLI_r1+Split
Accuracy
0.45
0.4
0.35
0.3
0.25
0
0.5
1
2
3
4
5
No. of samples augmented(x1000)


Performance of ELECTRA-small
ANLI_r2
ANLI_r2+Split
Accuracy
0.45
0.4
0.35
0.3
0.25
0
0.5
1
2
3
4
5
No. of samples augmented(x1000)


Performance of ELECTRA-small
ANLI_r3
ANLI_r3+Split
Accuracy
0.45
0.4
0.35
0.3
0.25
0
0.5
1
2
3
4
5
No. of samples augmented(x1000)


Performance of ELECTRA-small
HANS
HANS+Split
Accuracy
0.7
0.65
0.6
0.55
0.5
0.45
0
0.5
1
2
3
4
5
No. of samples augmented(x1000)


Performance of ELECTRA-small
Accuracy
0.7
0.65
0.6
0.55
0.5
0.45
0
0.5
1
2
3
4
5
MultiNLI
MultiNLI+Split
No. of samples augmented(x1000)


Performance of ELECTRA-small
Accuracy
0.9
0.85
0.8
0.75
0.7
0
0.5
1
2
3
4
5
SNLI
SNLI+Split
No. of samples augmented(x1000)

4 Conclusion

In this work, we present a simple but effective data augmentation and preprocessing method to improve the generalization and robustness of the NLI model. Such method improves the model by solving the following three problems: (1) word overlap; (2) numerical reasoning; (3) length mismatched. The experimental results show that the proposed method achieves more than 12% on the HANS dataset and 6% to 9% on the ANLI dataset. The method helps prevent the heuristic search between the premise and hypothesis. Also, the preprocessing technique(*Split*) sheds the light on classifying the hypothesis with the paragraph-level premise while we only train the model on the sentence-level premises. The method can be effective by adding only 3% of the original training dataset.

References

1. Giampiccolo, D., Magnini, B., Dagan, I., Dolan, B.: The third PASCAL recognizing textual entailment challenge. In: Sekine, S., et al. (eds.) Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing. pp. 1–9. Association for Computational Linguistics, Prague (2007)
2. Bowman, S.R., Angeli, G., Potts, C., Manning, C.D.: A large annotated corpus for learning natural language inference. In: Màrquez, L., et al. (eds.) EMNLP 2015. pp. 632–642. Association for Computational Linguistics, Lisbon (2015). https://doi.org/10.18653/v1/D15-1075
3. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT 2019. pp. 4171–4186. Association for Computational Linguistics, Minneapolis (2019). https://doi.org/10.18653/v1/N19-1423
4. Liu, Y., et al.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
5. Yang, Z., et al.: Xlnet: Generalized autoregressive pretraining for language understanding. In: NeurIPS 2019 (2019)
6. Clark, K., Luong, M.T., Le, Q.V., Manning, C.D.: Electra: Pre-training text encoders as discriminators rather than generators. In: ICLR 2020 (2020)
7. Levesque, H.J.: On our best behaviour. Artif. Intell. 212, 27–35 (2014)
8. Papernot, N., et al.: Practical black-box attacks against machine learning. In: Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security (2016)
9. Nie, Y., Bansal, M.: Shortcut-stacked sentence encoders for multi-domain inference. In: Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP. pp. 41–45. Association for Computational Linguistics, Copenhagen (2017). https://doi.org/10.18653/v1/W17-5308
10. Naik, A., et al.: Stress test evaluation for natural language inference. In: COLING 2018. pp. 2340–2353. Association for Computational Linguistics, Santa Fe (2018)
11. Williams, A., Nangia, N., Bowman, S.: A broad-coverage challenge corpus for sentence understanding through inference. In: NAACL-HLT 2018. pp. 1112–1122. Association for Computational Linguistics (2018)

12. McCoy, R.T., Pavlick, E., Linzen, T.: Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. arXiv preprint arXiv:1902.01007 (2019)
13. Nie, Y., et al.: Adversarial nli: A new benchmark for natural language understanding. In: ACL 2020. Association for Computational Linguistics (2020)