Unpacking the Resilience of SNLI Contradiction Examples to Attacks

Chetan Verma, Archit Agarwal

Abstract

Pre-trained models excel on NLI benchmarks like SNLI and MultiNLI, but their true language understanding remains uncertain. Models trained only on hypotheses and labels achieve high accuracy, indicating reliance on dataset biases and spurious correlations. To explore this issue, we applied the Universal Adversarial Attack to examine the model's vulnerabilities. Our analysis revealed substantial drops in accuracy for the entailment and neutral classes, whereas the contradiction class exhibited a smaller decline. Fine-tuning the model on an augmented dataset with adversarial examples restored its performance to near-baseline levels for both the standard and challenge sets. Our findings highlight the value of adversarial triggers in identifying spurious correlations and improving robustness while providing insights into the resilience of the contradiction class to adversarial attacks.

1 Introduction

Natural Language Inference (NLI) is a foundational task in Natural Language Processing (NLP) that evaluate's a model's natural language understanding (NLU). It involves determining whether a hypothesis is true (Entailment), false (Contradiction) or cannot be determined (Neutral) given a premise (Dagan et al., 2006, 2013). This reasoning ability is critical for mimicking human understanding and supports a wide range of applications. Consequently, when models achieve high accuracy on this task, it is often claimed that they have strong NLU capabilities. However, recent research (Poliak et al., 2018; Gururangan et al., 2018) shows that models achieve high accuracy even when trained on hypothesis-only datasets. This suggests that models exploit spurious correlations and superficial patterns known as dataset artifacts to predict the correct label rather than genuinely understanding the language.

To investigate this phenomenon and explore ways to improve the model, we focused on the Stanford NLI (SNLI) dataset (Bowman et al., 2015), one of the most widely used benchmarks for NLI tasks. For our study, we selected Efficiently Learning an Encoder that Classifies Token Replacements Accurately (ELECTRA-small[1]) (Clark et al., 2020). This model retains the same architecture as Bidirectional Encoder Representations from Transformers (BERT) but incorporates an improved training method. Moreover, ELECTRA is computationally efficient, making it a practical alternative to larger, more resource-intensive models.

Adversarial datasets were chosen as a key tool for assessing robustness due to their flexibility and cross-dataset applicability. In particular, we explored a method used in a previous study to generate Universal Adversarial Triggers (Wallace et al., 2019) to create a testing dataset and evaluated the robustness of ELECTRA. These triggers are transferable across models for all datasets, and unlike other adversarial attacks, they are context-independent and hence provide new insights into the general input-output patterns learned by the model.

To address the identified gaps and enhance the model's robustness, we fine-tuned it using a small, augmented training dataset (Liu et al., 2019). A detailed discussion of our methodology and analysis is provided later in the paper.

2 Background and Related Work

The evaluation and enhancement of machine learning models' robustness has emerged as a critical focus in recent research. A variety of techniques have been developed to create challenge

[1] We will use ELECTRA as an alias to refer to ELECTRA-small throughout this paper

sets that expose model vulnerabilities, with notable approaches including: (1) Contrast Sets: These are manually crafted modifications to test data that introduce small, label-changing alterations while maintaining lexical and syntactic integrity (Gardner et al., 2020); (2) Checklist Sets: This approach involves a systematic, task-agnostic framework for testing NLP models. Drawing inspiration from behavioral testing principles in software engineering, it employs diverse test types to evaluate model performance (Ribeiro et al., 2020); (3) Adversarial Challenge Sets: These datasets are deliberately modified to provoke problematic outputs, revealing critical weaknesses in models (Jia and Liang, 2017; Wallace et al., 2019).

To address the vulnerabilities identified through such challenge sets, researchers have proposed and employed several mitigation techniques, including: (1) Adversarial Data Training: This approach incorporates challenge sets directly into the training process or employs adversarial data augmentation to strengthen model performance against adversarial inputs (Liu et al., 2019); (2) Ensemble-based debiasing: Involves learning a biased model that captures dataset-specific clues, and then training a debiased model on the residual errors (He et al., 2019)

These techniques collectively represent significant strides in fortifying machine learning models against potential vulnerabilities, ensuring more reliable and robust performance in diverse applications.

3 Methodology

3.1 Universal Adversarial Attack

Universal adversarial triggers are generated tokens designed to manipulate a model's predictions. When appended to the beginning or end of an input, these triggers can compel the model to produce a prediction that deviates from the gold label.

Why Universal Triggers ? (1) Black-Box Capability: These triggers can be generated without any access to the target model, making them effective even in black-box scenarios; (2) Model Transferability: These triggers are universal, that is, they are capable of attacking and transferring across different models; and (3) Context Independence: Their independence from context provides valuable insights into the general input-output patterns of the model. Table-1 illustrates the impact of universal adversarial triggers in altering predicted labels, highlighting their effectiveness in manipulating model outputs.

3.1.1 Attack Equation

Given a model f (with white-box access), a text input composed of tokens t (which could represent words, sub-words, or characters), and a target label $\tilde{y}$, the goal is to generate triggers t_{adv} to append to the front or back of the input token t. In a non-universal adversarial setting, this can be mathematically expressed as:

$$f(t_{\text{adv}}; t) = \tilde{y} \tag{1}$$

For a universal adversarial setting, the goal is to optimize the universal trigger such that the loss function for the target class $\tilde{y}$ is minimized across all inputs from a dataset. Mathematically, this can be represented as:

$$\arg\min_{t_{\text{adv}}} E_{t\sim T}\left[\mathcal{L}(\tilde{y}, f(t_{\text{adv}}; t))\right] \tag{2}$$

where T represents all input instances from the dataset, and $\mathcal{L}$ represents the loss function.

3.1.2 Trigger Search Algorithm

Next, we start by selecting a trigger length: longer triggers tend to be effective, whereas shorter triggers are less noticeable. To initialize trigger creation, we prepend a simple token such as the character *a* or word *the* to the beginning of all inputs.

We incrementally refine the tokens in the trigger to optimize the loss function associated with target prediction, leveraging a technique inspired by HotFlip (Ebrahimi et al., 2018), that uses the token's gradient to get the token replacement. To use this technique, the trigger token t_{adv} is represented as one-hot vectors and embedded to form e_{adv}.

The HotFlip-inspired technique uses a linear approximation of the task loss. Specifically we update the embeddings for each trigger token e_{adv_i} to minimize the loss by applying a first-order Taylor approximation around the current token embedding:

$$\arg\min_{e'_i \in \mathcal{V}} \left[e'_i - e_{\text{adv}_i}\right]^T \nabla e_{\text{adv}_i}\mathcal{L} \tag{3}$$

where $\mathcal{V}$ is the set of all token embeddings in the model's vocabulary, and $\nabla e_{\text{adv}_i}\mathcal{L}$ is the average of the task loss over a batch. For our use-case, NLI (a classification task), we use the cross-entropy loss to optimize the attack.

Gold Label	Input (Premise and Hypothesis)	Predicted Label
Entailment (0)	**Premise:** A woman in a purple shirt buying food from a street vendor. **Hypothesis:** nobody a woman makes a purchase from a vendor outside.	0 → 2
Neutral (1)	**Premise:** A youth is kicking a soccer ball in an empty brick area. **Hypothesis:** cats a funny human kicking.	1 → 2
Contradiction (2)	**Premise:** A person dressed in red and black outside a cracked wall. **Hypothesis:** anxiously a person in red and black falling through the ionosphere.	2 → 1

Table 1: Effect of universal triggers (highlighted in red) on predicted labels across three classes

	Universal			Random		
Targeted Class	**Trigger**	**Majority Class**	**Score**	**Trigger**	**Majority Class**	**Score**
Entailment (0)	nobody	2	0.96	diners	1	0.47
	no	2	0.83	hands	2	0.38
Neutral (1)	cats	2	0.96	road	1	0.40
	cat	2	0.85	mass	0	0.38
Contradiction (2)	joyously	1	1.00	remain	1	0.62
	celebrating	1	0.79	rose	1	0.59

Table 2: Comparison of Universal and Random Triggers

3.2 Inoculation by Fine-Tuning

To address the identified vulnerabilities, we employed the *Inoculation by Fine-Tuning* technique (Liu et al., 2019). This method involves fine-tuning a pre-trained model on a small, carefully designed training dataset. Following fine-tuning, the model typically exhibits one of three behaviors:

1. **Reduced Performance Gap**: The performance disparity between the original test set and the challenge set decreases, with the model maintaining strong performance across both datasets. This outcome suggests that the observed gap originates from the dataset itself rather than inherent limitations of the model.

2. **Unchanged Performance**: The model's performance remains static, indicating an inability to adapt to the challenge set. This points to potential limitations within the model's architecture or design as the root cause.

3. **Decreased Performance**: The model's performance on the original dataset declines, even if improvements are observed on the challenge set. This behavior indicates potential overfitting to the adversarial examples introduced during fine-tuning, rather than addressing the underlying issue.

By applying this technique, we effectively diagnosed and mitigated the identified issues, strengthening the system's resilience and addressing key vulnerabilities.

4 Experiments

This section describes the process of generating triggers (a word in this case), creating attacks using these triggers, and subsequently training and evaluating the ELECTRA model to assess its ability to learn and perform the underlying NLI task effectively.

4.1 Generation of Triggers

4.1.1 Universal Triggers

Universal triggers are created using the Universal Adversarial Attack (Section-3.1). Initially, the token representing the word *the* is prepended to the targeted examples and then trigger search algorithm (Section-3.1.2) is applied to generate the universal triggers. These triggers are derived using the Enhanced Sequential Inference Model (ESIM) (Chen et al., 2017) with GloVe embeddings (Pennington et al., 2014). To simulate a realistic scenario, the generated triggers are tested using the ELECTRA model, assuming black-box access. This setup mimics real-world conditions, where white-box access to deployed models is typically unavailable, but testing their robustness is still necessary.

4.1.2 Random Triggers

As a baseline for the Universal Adversarial Attack, triggers are generated using a Random Attack approach. In this method, words are randomly selected from the SNLI dataset's vocabulary, ensuring a uniform distribution across all three classes. These selected words are then prepended to the hypotheses of SNLI examples to create the random attack.

4.1.3 Examples and Correlation Score

Table-2 presents examples of both universal and random triggers, along with their corresponding majority class (the class in which the trigger appears most frequently) and correlation score. The correlation score is defined as the conditional probability of a label l given a word w, and it is mathematically expressed as:

$$p(l|w) = \frac{count(w,l)}{count(w)} \tag{4}$$

4.2 Challenge Sets and Trigger-Augmented Dataset

To systematically evaluate model performance and mitigate reliance on spurious correlations, we developed two challenge sets and a Trigger-Augmented dataset. The challenge sets assess the model's robustness under adversarial conditions, while the Trigger-Augmented dataset aims to enhance generalization by addressing dataset-specific biases. The construction and purpose of these datasets are described below:

1. **Challenge Set I** (Validation split with universal triggers[2]): This set evaluates the model's robustness and understanding of the core NLI task. It consists of 1,000 examples, randomly sampled from each label class in the validation split of the SNLI dataset. Universal triggers (detailed in Section-4.1.1) are prepended to the hypothesis in these examples. This setup enables a focused evaluation of the model's performance across all label classes in the presence of adversarial triggers.

2. **Challenge Set II** (Validation split with random triggers[3]): Designed as a baseline for comparison with universal triggers, this set follows the same construction process as Challenge Set I but replaces universal triggers with random triggers (detailed in Section-4.1.2). This set provides a point of reference for measuring the impact of universal triggers on model performance by isolating the effect of non-specific, randomly chosen triggers.

3. **Trigger-Augmented Dataset** (Train split with universal triggers): To reduce the model's reliance on spurious correlations present in the original SNLI dataset, a fine-tuning dataset was created. This dataset contains 6,000 training examples, with 3,000 left unmodified and the remaining 3,000 modified by prepending universal triggers to their hypothesis. This augmentation encourages the model to prioritize semantically meaningful features over spurious patterns during training (refer to Section-3.2).

By utilizing these datasets, we aim to systematically evaluate and enhance the robustness of the ELECTRA model under both adversarial and standard conditions.

4.3 Training and Evaluation process

The ELECTRA model[4] was trained and fine-tuned on a single machine equipped with an NVIDIA T4 GPU. The training process utilized the HuggingFace Trainer framework, configured with a maximum sequence length of 128 to ensure that over 96% of examples from the SNLI dataset were fully captured without truncation. A batch size of 256 was selected, while all other parameters were left at their default settings.

[2] Our database for universal triggers can be found at `https://huggingface.co/datasets/ckverma/snli_universal`

[3] Our database for random triggers can be found at `https://huggingface.co/datasets/ckverma/snli_random`

[4] Our code for training the model can be found at `https://github.com/ckvermaAI/SNLI-Attack-Analysis.git`

Training was conducted in two stages. First, the model was trained on the original SNLI dataset for three epochs to establish a strong foundational understanding of natural language inference. This was followed by a fine-tuning phase, where the model was adapted using the Trigger-Augmented dataset to enhance robustness and mitigate reliance on spurious correlations. This two-stage process allowed the model to effectively balance learning from the original data and adapting to the additional challenge sets.

5 Results and Analysis

5.1 Training ELECTRA on the SNLI Dataset

The Electra model was trained on the SNLI dataset for three epochs, achieving a validation accuracy of 88.98%. During this initial phase, evaluations were performed on three datasets: the validation subset (comprising 1,000 randomly sampled examples from the SNLI validation split), Challenge Set I, and Challenge Set II. The results from this stage are summarized in the first three rows of Table-3.

5.2 Fine-Tuning ELECTRA model

In the second phase, the model underwent fine-tuning on the Trigger-Augmented dataset for one epoch. This step was designed to reduce the model's dependence on spurious correlations present in the original SNLI dataset, thereby enhancing its robustness. Post-fine-tuning, the model was re-evaluated on the validation subset and Challenge Set I, with the results documented in the last two rows of Table-3.

5.3 Analyzing the Results

5.3.1 Effectiveness of Triggers

Table-4 highlights the impact of universal and random triggers. Universal triggers effectively alter the model's predictions for entailment and neutral examples, often shifting them to other classes. In contrast, random triggers have minimal influence, affecting approximately 20% of the entailment examples. These findings demonstrate the superior efficacy of universal triggers in manipulating model predictions (compared to random triggers).

5.3.2 Success of Universal Triggers

The Universal Adversarial Attack generates triggers that are strongly correlated with a competing class (the class other than the intended target). Table-2 highlights the high correlation scores between the universal triggers and their associated dominant (or majority) class. When these triggers are appended to SNLI examples from the targeted class, they exploit the model's reliance on spurious correlations, leading it to favor the competing class over the intended target. For example, the universal trigger *nobody* is closely associated with the contradiction class. When prepended to examples from the entailment class, it causes the model to misclassify them as contradictions (1st row in Table-1).

5.3.3 De-biasing the Model

The uniform distribution of universal triggers in the Trigger-Augmented dataset helps the model unlearn spurious correlations present in the original SNLI dataset. The two-stage training process balances foundational learning from the original data while mitigating biases using the Trigger-Augmented dataset. As shown in Table-3 (1st, 2nd and, 5th row), this approach significantly enhances the model's overall performance and resilience (as highlighted in Section-3.2).

5.3.4 Decoding Attacks on the Contradiction Class

The contradiction class contains more correlated words in comparison to the entailment and neutral classes. The cumulative frequency of the top five correlated words is 312 for contradictions, 128 for neutral, and 57 for entailment. This abundance of correlated words makes contradictions particularly vulnerable. However, flipping predictions for contradiction-class examples to entailment or neutral by simply prepending tokens is feasible only if the example lacks these giveaway words. This is the reason why ELECTRA model's ability to correctly predict contradiction examples reduces by only 7.43% with the introduction of universal triggers (refer to the results in Table-3).

6 Conclusion and Future work

In this study, we systematically investigated the vulnerabilities and biases in NLI models, proposing methods to enhance their robustness. Our findings demonstrated the effectiveness of universal triggers in exploiting spurious correlations to manipulate NLI model predictions, significantly outperforming random triggers. Moreover, Trigger-Augmented training proved successful in mitigat-

Dataset	**Triggers (Model)**	**Entailment (%)**	**Neutral (%)**	**Contradiction (%)**
Validation Subset	Original (Pre-Finetune)	90.23	86.70	91.06
Challenge Set I	Universal (Pre-Finetune)	25.78	25.76	83.63
Challenge Set II	Random (Pre-Finetune)	71.20	83.98	91.57
Validation Subset	Original (Post-Finetune)	88.92	84.81	91.16
Challenge Set I	Universal (Post-Finetune)	90.13	87.53	91.96

Table 3: Performance Summary of ELECTRA on Validation subset of SNLI dataset and Challenge Sets (I and II)

Ground Truth	**Data**	**E%**	**N%**	**C%**
Entailment	Validation subset	**90.23**	7.65	2.11
	Challenge Set I	25.78	32.93	**41.29**
	Challenge Set II	**71.20**	21.85	6.95
Neutral	Validation subset	7.33	**86.70**	5.97
	Challenge Set I	0.63	25.76	**73.61**
	Challenge Set II	5.97	**83.98**	10.05
Contradiction	Validation subset	1.31	7.63	**91.06**
	Challenge Set I	0.60	15.76	**83.63**
	Challenge Set II	1.41	7.03	**91.57**

Table 4: ELECTRA model's prediction distribution for different datasets. Each row shows a particular dataset and each column shows how often model predicts a particular class. For example, on the challenge set I, neutral examples are classified as contradiction examples 73.61% times.

ing biases, thereby improving model resilience. This approach also underscored the nuanced challenges associated with attacking the contradiction class, shedding light on areas requiring further exploration.

For future work, we aim to explore diverse attack strategies (Song et al., 2021) beyond merely prepending triggers to hypotheses. Such strategies will help uncover additional weaknesses in NLI datasets, providing deeper insights into designing more robust datasets and improving the training processes for NLI models.

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. *CoRR*, abs/1508.05326.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced lstm for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pretraining text encoders as discriminators rather than generators. *CoRR*, abs/2003.10555.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognising textual entailment challenge. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 177–190, Berlin, Heidelberg. Springer Berlin Heidelberg.

Ido Dagan, Dan Roth, Mark Sammons, and Fabio Zanzotto. 2013. Recognizing textual entailment: Models and applications. *Synthesis Lectures on Human Language Technologies*, 6(4):1–222. Publisher Copyright: © Morgan and Claypool Publishers. All rights reserved.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. HotFlip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36, Melbourne, Australia. Association for Computational Linguistics.

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. 2020. Evaluating models' local decision boundaries via contrast sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1307–1323, Online. Association for Computational Linguistics.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R. Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. *CoRR*, abs/1803.02324.

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

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Nelson F. Liu, Roy Schwartz, and Noah A. Smith. 2019. Inoculation by fine-tuning: A method for analyzing challenge datasets. *CoRR*, abs/1904.02668.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. *CoRR*, abs/1805.01042.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

Liwei Song, Xinwei Yu, Hsuan-Tung Peng, and Karthik Narasimhan. 2021. Universal adversarial attacks with natural triggers for text classification.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for NLP. *CoRR*, abs/1908.07125.