Measuring and Improving Adversarial Robustness in Text Classifiers using Robustness Budget Curves

Ron Alex
University of Houston
Houston, Texas

December 6, 2025

Abstract

Transformer-based natural language architectures like DistilBERT achieve state-of-the-art performance on benchmarks but are extremely brittle to small-magnitude input perturbations. This paper will investigate the vulnerability of DistilBERT-base-uncased against three attack recipes from the TextAttack framework: TextFooler, TextBugger, and PWWS. I evaluated the robustness across two distinct text modalities: Short-sequence sentiment classification using the Stanford Sentiment Tree V2 dataset and long-document classification using the IMDb dataset. I evaluated a layered defense strategy combining adversarial training and inference-time input sanitation. My results demonstrate that while single-layer defenses have specific failure points, such as adversarial training struggling with lexical shift attacks, and input sanitation struggling against semantic shift attacks, a combined architecture can defend against multiple different modalities of input perturbation, yielding an overall gain in robustness. Furthermore, I identified a correlation between input sequence length and attack success, giving evidence to the 'Surface Area Problem,' where longer documents provide an exponentially larger search space that renders standard defenses less effective.

1 Introduction

Since 2019, there have been significant advancements in the field of Natural Language Processing using deep learning algorithms. Many state-of-the-art solutions have already reached or even surpassed human-level baselines in some tasks, and as such, are now in practical use everywhere, from text classification, machine translation, malware detection, etc (Khurana et al., 2017). Using deep neural networks, an NLP pipeline learns word representation in text and contextual details in sentences, and this text modeling can be utilized using Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), or more recently, Transformer based learning models. However, despite exceptional accuracy, the interpretability of deep neural networks is still unsatisfactory, as they essentially work as black boxes, and it is difficult to understand what exactly a model has learned. A problem that arises from this poor interpretability is evaluating the robustness of deep neural networks.

Recent works have used imperceptible perturbations to evaluate the robustness of deep neural networks and found that they are not robust to these perturbations. Papers such as (Szegedy et al., 2013), (Goodfellow et al., 2014) have popularized this research topic, giving rise to the field of Adversarial Machine Learning. More recently, there have been a number of proposed adversarial attacks for deep neural networks, which raise concerns about the robustness of state-of-the-art models. Adversarial attacks present a significant security threat to the practical deployment of NLP models, such as in spam detection, content moderation, etc. Attacks can take various forms, such as lexical-level shifts, synonym-swap attacks, sentence-level attacks, and multi-level attacks, which can include a mixture of many different attacks (Zhang et al., 2019).

As such, the importance of adversarial defense is self-evident and has become a growing field with a large amount of work done. Adversarial defense strategies in NLP can be broadly classified into three fields: adversarial training-based, perturbation control-based, and certification-based methods. Of these, adversarial-training-based defenses remain the most widely used, which attempt to regularize the model's

Dataset	Avg. Length	Train size	Validation size	Test size
SST-2	∼54	∼65,000	872	∼2,000
IMDb	∼1,300	∼20,000	∼5,000	∼25,000

Table 1: Dataset Statistics across 2 datasets

decision boundary during learning (Zhang et al., 2019). Few studies have systematically examined the effects that the layering of multiple defense architectures has on a transformer model.

In this paper, I will conduct an evaluation on how transformer models' robustness can be improved using a layered defense strategy, with inference-time text-sanitization in addition to adversarial training. I will be using DistilBERT, a distilled Transformer model, fine-tuned on the Stanford Sentiment Tree V2 (SST-2) and IMDb datasets. My contributions are as follows.

1. I quantify the degradation of Transformer performance under varying perturbation budget constraints (5%, 10%, 20%).
2. I demonstrate that a layered defense strategy can outperform individual strategies.
3. I provide evidence for the 'Attack Surface Problem,' where larger text sequences are inherently more difficult to defend.

2 Methodology

2.1 Datasets and preprocessing

Datasets: To evaluate the robustness across different text modalities, I selected two datasets representing opposite ends of the sequence length spectrum, both from HuggingFace:

- SST-2 (Stanford Sentiment Treebank V2): Introduced in (Socher et al., 2013), SST-2 is a binary sentiment classification dataset that contains single sentences extracted from movie reviews. This represents short-text classification, which could be found in social media or product reviews. The average sequence length of this dataset is ˜54 characters.
- IMDb Movie reviews: Introduced in (Maas et al., 2011), the IMDb dataset is a binary sentiment classification dataset of highly polarized reviews compiled from the movie review site www.imdb.com. These represent long-sequence text classification, and the average sequence length of this dataset is ˜1300 characters.

Data Splitting: As these datasets are popular and used as benchmarks for competitions, they lack public labels or have inconsistent validation splits. To ensure uniform evaluation, I constructed custom splits:

- SST-2: As the original dataset lacked a labeled test split, I stratified and extracted 2,000 random instances from the training split to serve as the test split.
- IMDb: The default validation split was unsupervised, which meant that it had no associated labels. I extracted 5,000 instances from the training split to create a supervised training split.
- Table 1 presents dataset statistics, including class distribution and average sequence length.

Preprocessing: All text was normalized to lowercase, and every instance under 5 characters was removed. Because of differing column labels, both datasets were standardized to have uniform column headers of 'text' and 'label.'

2.2 Threat Model

I utilized the TextAttack framework, as put forward in (Morris et al., 2020), to create three distinct adversarial strategies:

- TextFooler (Jin, Jin, Joey Tianyi Zhou, et al., 2019): A semantic attack that replaces words with synonyms that have high cosine similarity in the word embedding space. It targets the model's semantic understanding.
- TextBugger (Li et al., 2019): A hybrid attack that employs both word swaps and character-level noise (insertions, deletions, neighbor swaps). It exploits the model's tokenization fragility.
- PWWS (Ren et al., 2019): A greedy attack using Probability Weighted Word Saliency to determine which words have the highest impact on the classification layer.

I constrained these attacks with varying Perturbation Budgets ($\epsilon \in \{0.05, 0.10, 0.20\}$) to simulate attackers with varying degrees of freedom.

2.3 Defense Architecture

Defense 1 - Adversarial Training: Adversarial Training functions as a form of data augmentation, injecting "hard" examples into the training distribution.

- Generation: I generated 2,500 successful adversarial examples for SST-2 and 900 for IMDb using the attacks described above.
- Augmentation: These examples were added to the original clean training data (preserving the original ground-truth labels) to create a dataset containing 10% adversarial samples.
- Training: The model was fine-tuned on this augmented dataset for 3 epochs.

Defense 2 - Input Sanitation: I developed a caching-optimized Input Sanitizer based on TextBlob. This defense operates at inference time, intercepting the input string before tokenization..

- Mechanism: The sanitizer calculates the edit distance of input words against a standard lexicon. Words identified as Out-Of-Vocabulary (OOV) typos (e.g., "*mvie*") are rectified to their nearest valid neighbor (e.g., "*movie*").
- Optimization: Due to the computational cost of spell-checking long IMDb documents, I implemented a memoization (caching) layer to store previously corrected tokens..

Defense 3 - Combined Approach: I hypothesize that Adversarial Training and Sanitation cover different vulnerabilities. I combined them by wrapping the Adversarially Trained Model inside the Input Sanitizer.

3 Experimental Setup

All models were trained and attacked using the Hugging Face Trainer API on an NVIDIA T4 GPU using Google Colaboratory. I used DistilBERT-base-uncased as the target model, as it had a low computational overhead with only 66 million parameters, and displayed extremely high accuracy in sentiment classification tasks. I used a batch size of 16 with floating-point-precision-16. The models are optimized with a weight decay of 0.01 and a learning rate of $2e^{-5}$ to ensure time for convergence. Baseline models were trained for 2 epochs, and Robust models for 3, to ensure low computational overhead and to reduce overfitting.

Prior to attacks, our baseline models achieved:

- IMDb Clean Accuracy: 92.67%
- SST-2 Clean Accuracy: 90.67%

4 Results

Tables 2 compare the performance of the 3 different attack methods: TextFooler, TextBugger, and PWWS in terms of 2 evaluation metrics: Accuracy Under Attack(AUA), and the Average number of queries per attack.
For detailed plots of the robustness budget curves, please refer to Appendix A.1.

Attack	Dataset	Clean %	AUA% ↑			Avg. Queries		
			$\epsilon = 0.05$	$\epsilon = 0.1$	$\epsilon = 0.2$	$\epsilon = 0.05$	$\epsilon = 0.1$	$\epsilon = 0.2$
Baseline Models								
TextFooler	SST-2	90.67	47.33	32.33	16.67	41	52	63
	IMDb	92.00	24.67	13.67	5.00	264	297	327
TextBugger	SST-2	90.67	48.67	39.67	29.00	24	29	34
	IMDb	92.00	40.00	23.00	12.00	218	265	301
PWWS	SST-2	90.67	41.33	30.67	19.33	104	108	113
	IMDb	92.67	24.00	12.00	2.00	1408	1430	1446
Adversarial Training								
TextFooler	SST-2	94.00	64.00	48.00	21.00	44	56	75
	IMDb	91.00	38.00	26.00	10.00	413	489	567
TextBugger	SST-2	94.00	44.00	29.00	23.00	26	29	32
	IMDb	91.00	47.00	20.00	12.00	296	378	425
PWWS	SST-2	94.00	63.00	46.00	30.00	99	105	114
	IMDb	91.00	40.00	25.00	14.00	1532	1590	1641
Combined Defense								
TextFooler	SST-2	92.00	54.00	39.00	19.00	43	53	65
	IMDb	90.00	42.00	20.00	10.00	527	599	653
TextBugger	SST-2	92.00	63.00	50.00	40.00	25	29	35
	IMDb	90.00	48.00	38.00	26.00	326	394	473
PWWS	SST-2	92.00	56.00	39.00	23.00	97	102	109
	IMDb	90.00	32.00	22.00	12.00	1480	1518	1556

Table 2: Comprehensive comparison of AUA and query counts across Baseline, Adversarial, and Combined models.

4.1 Baseline Analysis

The Baseline Models section in Table 2 reveals a higher decrease in AUA for the model fine-tuned on IMDb compared to SST-2, with a minimum AUA of only 2 under PWWS for IMDb at $\epsilon = .2$, compared to a minimum AUA of 16.67 for SST-2 under TextFooler $\epsilon = .2$. This is evidence of our assumption that larger text sequences are more vulnerable to input perturbation compared to shorter text sequences.

4.2 Adversarial Defense Analysis

The Adversarial Training section in Table 2 shows robustness after adversarial training. I noticeably improved AUA at nearly all attacks and budgets, but AUA for TextBugger has actually dropped compared to baseline, even after adversarial training. I can attribute this to the way TextBugger generates attacks, as it uses lexical-level shifts that adversarial training cannot generalize to. This is evidence that single-layer defenses alone cannot defend against a wide range of adversarial attacks.

4.3 Combined Architecture Analysis

The Combined Defense section in Table 2 shows robustness after adding an input sanitation layer on top of an adversarial-trained model. I see improved AUA for all attacks and attack budgets for both datasets, with significant increases at $\epsilon = .05$ compared to baseline for SST-2 and IMDb. But the tables reveal that AUA for IMDb lags much further behind SST-2 for all defense architectures.

5 Discussion & Analysis

5.1 Weaknesses of Adversarial Training

While adversarial training proved highly effective against TextFooler and PWWS, it struggled against TextBugger on SST-2 (difference of -6% AUA compared to baseline). I can attribute this to tokenization artifacts. When TextBugger introduces a lexical shift, e.g, '*awesome*' to '*awsome*', the tokenizer, which is based on the WordPiece tokenization algorithm, splits the word into fragmented sub tokens (*aw*, *##some*) that the model has never seen in that context. This alters what the word is supposed to mean and creates noise. Adversarial training alone cannot combat this, as the model cannot learn a robust weight for every possible character permutation that TextBugger can generate.

5.2 The Efficacy of Layered Defense

The combined defense architecture achieved the highest AUA across all experiments. On SST-2, against TextBugger at $\epsilon = .05$, the combined model improved AUA from 49 to 63, an increase of 28.57%. The sanitizer fixed lexical noise, while the robust weights handled the remaining semantic shifts.

5.3 The Surface Area Problem (SST-2 vs. IMDb)

A critical finding of this paper is the discrepancy in defensibility between short and long texts. SST-2: Robustness gains were high (e.g., 15-20% improvement). IMDb: Robustness gains were lower (e.g., 5-10% improvement).

I attribute this to the Search Space Surface Area. In a short sentence (15 words), fixing one adversarial typo often disrupts the entire attack algorithm. In an IMDb review (200 words), fixing a typo only forces the attacker to pivot. With hundreds of words available to perturb, the attacker can easily find synonym swaps that bypass the sanitizer. This suggests that current defense methods scale poorly with sequence length.

5.4 Computational Cost of Robustness

Although the combined defense did not strictly prevent all attacks on IMDb, it was able to significantly increase the Cost to Attack. Comparing baseline to the combined architecture, we can see that the average number of queries to find a successful attack increased by 24%, which takes more time and computational overhead for the attacker. This forces the attacker to degrade the textual quality, rendering the adversarial perturbations increasingly perceptible to a human reader.

6 Limitations

Latency: The text input sanitation layer introduces latency. While the caching mechanism minimized this, real-time deployment would require optimized implementations rather than Python-based TextBlob.
Sanitation: This paper only discusses the efficacy of TextBlob, and does not go into detail about the more lightweight Regex-based text sanitation, or the heavier PySpellChecker-based text sanitation. .
Generalizability: My defenses were trained only on specific attack recipes. It remains unknown if these results can be concretely applied to other adversarial attack algorithms, such as genetic attacks.

7 Conclusion

This paper presents a systematic evaluation of adversarial robustness in transformer-based DistilBERT. I demonstrate that while Transformers are fragile to hybrid attacks, a layered defense strategy can significantly mitigate these vulnerabilities. We can conclude that:

- Input sanitation is essential for neutralizing character-level noise that breaks subword tokenizers.
- Adversarial training is essential for semantic robustness.

- Sequence length is a primary factor in vulnerability: longer sequences require more robust defenses.

Future work should focus on 'context-aware' sanitation that can detect adversarial synonyms based on sentence perplexity, addressing the limitations found in the IMDb experiments.

7.1 What I learned

Through this project, I learned that Transformer-based architectures like DistilBERT, despite their high performance on standard benchmarks, remain inherently brittle to adversarial perturbations, particularly character-level noise that fragments subword tokenization. A key realization was that single-layer defenses are insufficient for comprehensive security; Adversarial Training successfully mitigates semantic attacks (synonyms) but fails against lexical noise, whereas Input Sanitation neutralizes typos but cannot detect semantic shifts. Consequently, I learned that a layered defense strategy is essential to cover the full attack surface. Furthermore, the experiments revealed a critical correlation between sequence length and defensibility, which is evidence of the 'Surface Area Problem,' where longer documents (IMDb) proved significantly harder to defend than short sequences (SST-2) simply due to the larger search space available to the attacker. Ultimately, this demonstrates that achieving robustness requires not just model fine-tuning, but a holistic approach that increases the computational cost and textual degradation required for an attack to succeed.

References

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014, June 10). Generative Adversarial Networks. ArXiv.org. https://arxiv.org/abs/1406.2661

Goyal, S., Doddapaneni, S., Khapra, Mitesh M, & Ravindran, B. (2022). A Survey of Adversarial Defences and Robustness in NLP. ArXiv.org. https://arxiv.org/abs/2203.06414

Jin, D., Jin, Z., Zhou, J. T., & Szolovits, P. (2019). Is BERT Really Robust? A Strong Baseline for Natural Language Attack on Text Classification and Entailment. ArXiv.org. http://arxiv.org/abs/1907.11932

Khurana, D., Koli, A., Khatter, K., & Singh, S. (2017). Natural Language Processing: State of The Art, Current Trends and Challenges. ArXiv.org. https://arxiv.org/abs/1708.05148

Li, J., Ji, S., Du, T., Li, B., & Wang, T. (2019). TextBugger: Generating Adversarial Text Against Real-world Applications. Proceedings 2019 Network and Distributed System Security Symposium. https://doi.org/10.14722/ndss.2019.23138

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011, June 1). Learning Word Vectors for Sentiment Analysis. ACLWeb; Association for Computational Linguistics. https://aclanthology.org/P11-1015/

Morris, J. X., Lifland, E., Yoo, J. Y., Grigsby, J., Jin, D., & Qi, Y. (2020). TextAttack: A Framework for Adversarial Attacks, Data Augmentation, and Adversarial Training in NLP. ArXiv:2005.05909 [Cs]. https://arxiv.org/abs/2005.05909

Ren, S., Deng, Y., He, K., & Che, W. (2019, July 1). Generating Natural Language Adversarial Examples through Probability Weighted Word Saliency. ACLWeb; Association for Computational Linguistics. https://doi.org/10.18653/v1/P19-1103/

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. ArXiv.org. https://arxiv.org/abs/1910.01108

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., & Potts, C. (2013, October 1). Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. ACLWeb; Association for Computational Linguistics. https://aclanthology.org/D13-1170/

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2013). Intriguing properties of neural networks. ArXiv.org. https://arxiv.org/abs/1312.6199

TextAttack Documentation — TextAttack 0.3.9 documentation. (n.d.). Textattack.readthedocs.io. Retrieved December 5, 2025, from https://textattack.readthedocs.io/en/master/

Zhang, W., Sheng, Q. Z., Ahoud Alhazmi, & Li, C. (2019). Adversarial Attacks on Deep Learning Models in Natural Language Processing: A Survey. https://doi.org/10.48550/arxiv.1901.06796

A Appendix

A.1 Robustness Budget Curves

Per the discussion in Section 4, I provide the full robustness budget curves for the combined defense architecture below.

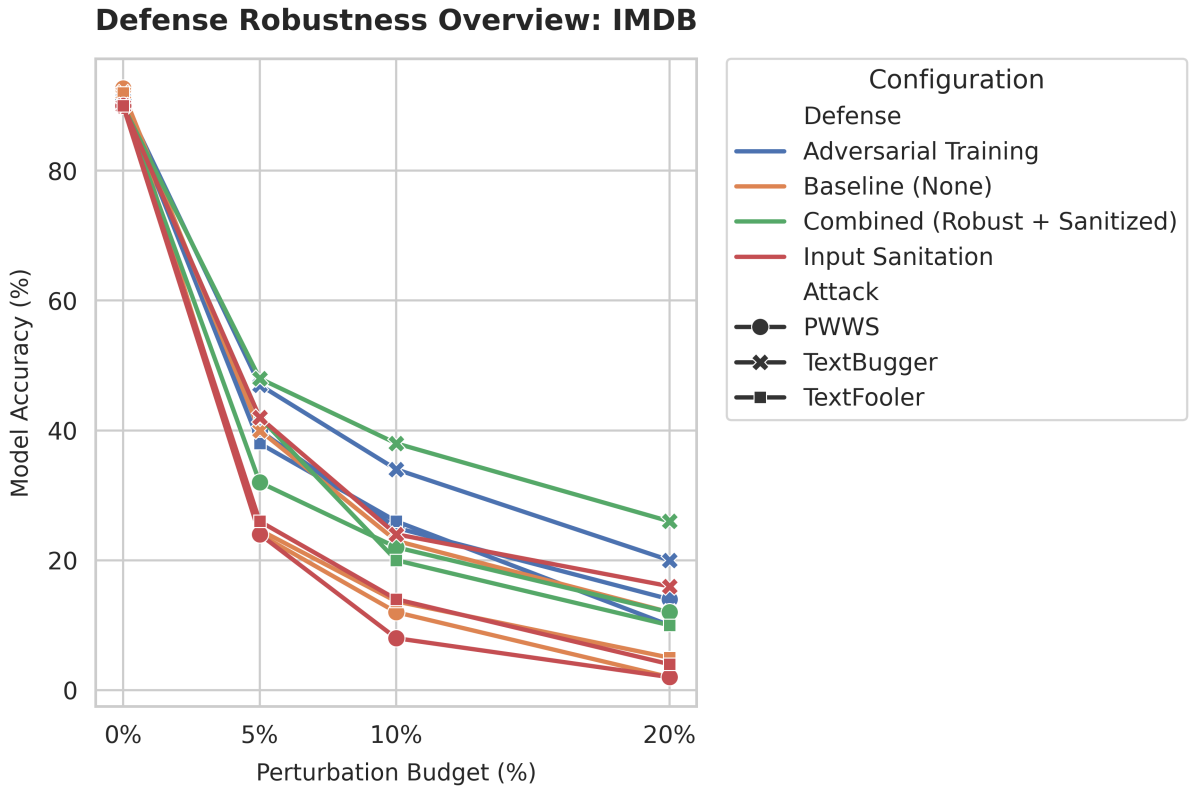

Figure 1: Detailed Robustness overview for IMDb.

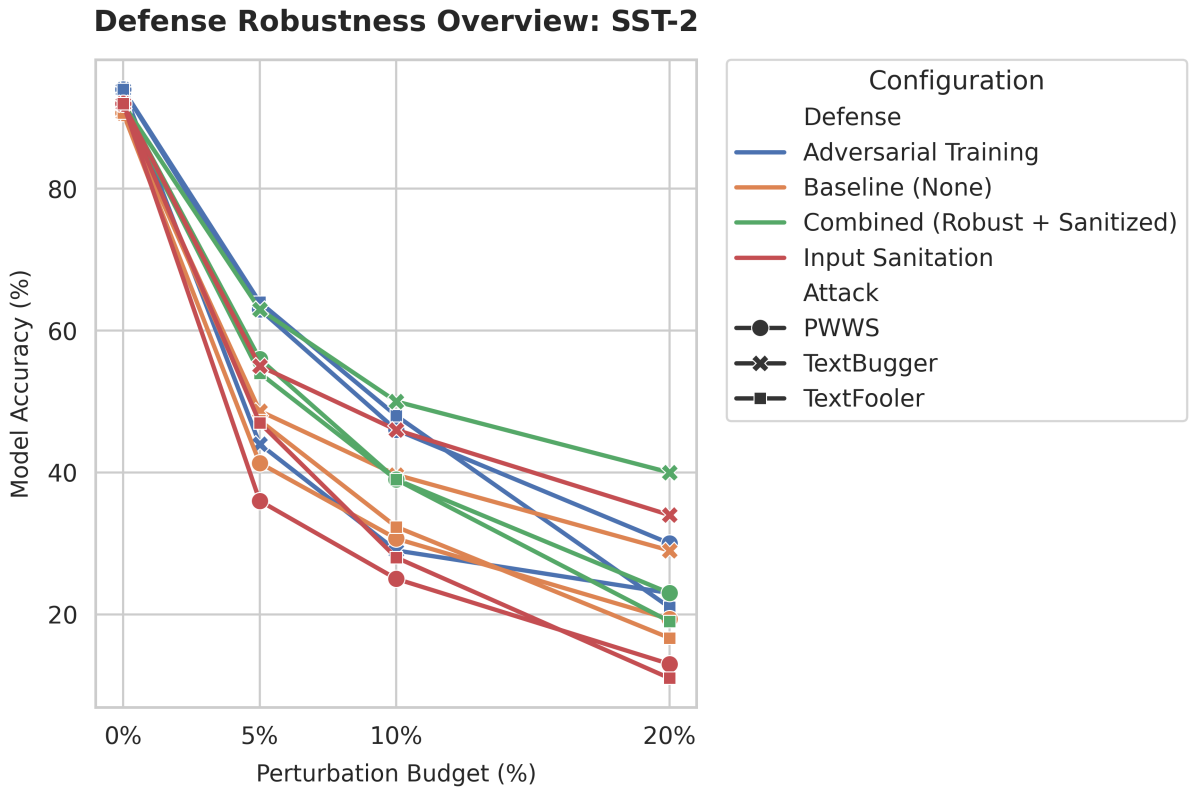

Figure 2: Detailed Robustness overview for SST-2.