Adversarial Attacks on NLP Models (Text Perturbation)

Nidhishree Talastha
talan01@pfw.edu
Purdue University Fort Wayne

Abstract

This research investigates the vulnerability of transformer-based NLP models to adversarial text perturbation attacks that preserve human readability while causing misclassification. I systematically evaluate four distinct attack methodologies—character-level manipulations, word-level substitutions, homoglyph replacements, and DeepWordBug techniques—against BERT and DistilBERT models fine-tuned on the IMDB dataset. Results reveal word-level and homoglyph attacks consistently achieved significantly higher success rates across both models, while requiring minimal text modifications. Notably, DistilBERT showed increased vulnerability to character-level attacks compared to BERT. These findings highlight fundamental architectural vulnerabilities in transformer models and emphasize the need for multi-layered defense strategies in security-critical NLP applications. Code for reproducing experiments is available at:https://github.com/nidhitalastha/CS59000-06_term_project

1 Introduction

Natural Language Processing models have become integral to various applications, including sentiment analysis and content moderation systems. Despite their impressive performance on standard benchmarks, these models exhibit concerning vulnerabilities to adversarial attacks. This research explores how transformer-based models respond to text perturbation attacks that preserve human readability while causing misclassification. By testing both BERT and DistilBERT architectures, the study investigates whether model size influences robustness against adversarial manipulation. The implementation leverages the TextAttack framework, which provides a modular design for constructing adversarial attacks through four components: goal functions, constraints, transformations, and search methods. Four distinct attack strategies are evaluated: character-level manipulations, word-level substitutions, homoglyph replacements, and DeepWordBug techniques. Understanding these vulnerabilities is essential for developing more robust NLP systems that maintain reliability in real-world deployments where adversarial manipulation poses genuine threats to security-critical applications.

2 Motivation

The motivation for this research stems from the growing security and fairness concerns regarding NLP models in real-world applications. As AI-driven systems increasingly power automated decision-making processes, their vulnerability to adversarial attacks poses serious ethical and technical challenges. Social media platforms, content moderation systems, and recommendation algorithms rely heavily on transformer-based models like BERT and DistilBERT to filter inappropriate content and analyze sentiment. However, adversaries can potentially exploit these models by making subtle text modifications that evade detection while preserving harmful intent. For instance, offensive comments could be slightly altered to bypass hate speech detection, or product reviews could be manipulated to falsely influence sentiment analysis systems. By implementing and evaluating multiple attack methodologies through the TextAttack framework, this research aims to quantify vulnerability patterns across model architectures and contribute to the development of more robust AI systems that can resist manipulation while maintaining high accuracy and fairness in critical applications.

3 Related Work

Jin et al. [3] established that transformer-based models like BERT remain vulnerable to adversarial attacks through their TextFooler approach. Their re-

search demonstrated that semantically similar word substitutions could significantly degrade model performance while preserving the original text's meaning from a human perspective. This work highlighted the gap between human and machine language understanding that my research further explores through comparative analysis of BERT and DistilBERT models.

Gao et al. [2] introduced DeepWordBug, revealing that NLP models are susceptible to character-level perturbations despite their sophisticated architectures. Their findings showed that even minor alterations to text could cause significant classification errors, indicating fundamental weaknesses in how models process language at the character level. My research extends this understanding by evaluating character-level attack effectiveness across different model sizes.

Eger et al. [1] explored homoglyph attacks against NLP systems, demonstrating that visual manipulation techniques could effectively deceive models while appearing unchanged to human readers. Their work exposed vulnerabilities in tokenization processes that my research investigates across both BERT and DistilBERT architectures to determine whether model size influences susceptibility to these types of attacks.

Morris et al. [4] developed TextAttack, a comprehensive framework for implementing adversarial attacks in NLP. Their contribution standardized the evaluation and comparison of different attack strategies through a modular architecture. This framework provides the foundation for my systematic comparison of vulnerability patterns across model sizes and attack types, offering valuable insights for developing more robust NLP systems.

4 Methodology

4.1 Dataset Preparation

The IMDB Movie Reviews dataset was employed for sentiment classification experiments. This dataset was loaded using the Hugging Face datasets library, providing 25,000 training samples and 25,000 test samples equally distributed between positive and negative sentiment classes. Basic text preprocessing was applied, including lowercase conversion, HTML tag removal, and special character filtering using regular expressions. For transformer models, the texts were tokenized using model-specific tokenizers (`BertTokenizer` and `DistilBertTokenizer` from the transformers library) with a maximum sequence length of 128 tokens. The encoded inputs included input IDs, attention masks, and corresponding labels converted to PyTorch tensors. The processed dataset was then prepared for model training by creating PyTorch `DataLoaders` with appropriate batch sizes using `RandomSampler` for training and `SequentialSampler` for validation.

4.2 Training Baseline Models

Two transformer-based models were fine-tuned: BERT (`bert-base-uncased`) and DistilBERT (`distilbert-base-uncased`) for sentiment classification. The training utilized the transformers library, with models initialized using the pre-trained weights. The `AdamW` optimizer with a learning rate of 2e-5 and epsilon of 1e-8 was employed, along with a linear learning rate scheduler with warmup. The training process included gradient clipping to prevent exploding gradients, setting the maximum gradient norm to 1.0. For BERT, the first epoch was selected as the optimal checkpoint as it demonstrated the best balance between accuracy and risk of overfitting. In contrast, DistilBERT required additional training, with the third epoch yielding the most robust performance. Model evaluation metrics included accuracy, precision, recall, and F1-score, calculated using the scikit-learn library.

4.3 Implementing Adversarial Attacks

Four distinct adversarial attack methodologies were implemented using the TextAttack framework [4]. A custom model wrapper class was created to adapt the fine-tuned PyTorch models for compatibility with TextAttack. The framework's modular design allowed for the implementation of attacks through four components: goal functions, constraints, transformations, and search methods. The `UntargetedClassification` goal function was used for all attacks, with common constraints including `RepeatModification` and `StopwordModification`. Four attack types were implemented:

1. Character-level attack (`WordSwapNeighboringCharacterSwap`): Manipulated text by swapping adjacent characters, with constraints limiting modifications to at most 20% of characters per word and a maximum of 20 words per example [1].

2. Word-level attack (`WordSwapEmbedding`): Replaced original words with semantically simi-

lar alternatives using counter-fitted GloVe embeddings, maintaining part-of-speech consistency with a semantic similarity threshold of 0.7 [3].

3. Homoglyph attack (`WordSwapHomoglyphSwap`): Exploited visual similarity by replacing standard Latin characters with their visually similar Unicode counterparts [1].

4. DeepWordBug attack: Employed a gradient-based scoring function to identify tokens critical to model prediction, applying character-level transformations [2].

The `GreedyWordSwapWIR` search method with "delete" as the importance ranking method was used to find optimal perturbations. Attack termination conditions included successful prediction change or reaching the maximum query limit (500 queries for character-level, 1000 for word-level attacks).

4.4 Visualization and Analysis

Custom visualization scripts were developed using Matplotlib and Seaborn to analyze attack effectiveness. These visualizations included attack success rates across different models and attack types, model performance degradation after adversarial attacks, and text perturbation examples. The `difflib` library was used to identify and highlight differences between original and perturbed texts. For perturbation impact analysis, a systematic examination of the position of perturbations in text was conducted, calculating relative positions of perturbed words and grouping them into positional buckets.

5 Experiments

5.1 Setup

All model training was conducted on Google Colab with a T4 GPU, while adversarial attacks were executed on the Kaggle platform utilizing a P100 GPU. A fixed random seed (42) was maintained throughout all experiments to ensure reproducibility. The BERT model was trained for 3 epochs, with checkpoints saved after each epoch. Based on validation performance, the first epoch checkpoint was selected as optimal for BERT as it demonstrated the best balance between accuracy and risk of overfitting. DistilBERT required additional training, with the model trained for 3 epochs and the third epoch checkpoint selected for attacks. Both models achieved comparable baseline performance with approximately 88% accuracy on the IMDB test set before adversarial testing.

5.2 Adversarial Testing

To ensure fair comparison between models, each attack was tested against both BERT and DistilBERT using an identical number of examples. The testing process followed an incremental approach, beginning with a preliminary evaluation using 10 examples to verify attack functionality and computational feasibility. Upon confirming proper implementation, the sample size was increased to 100 examples for more reliable preliminary results. Finally, full-scale testing was conducted with 1000 examples per attack methodology for statistically significant findings. Importantly, while the number of examples was consistent across models for each attack type, the test examples were not identical across different attack methodologies to prevent potential biases from specific examples.

5.3 Evaluation Metrics

The effectiveness of each adversarial attack was evaluated using several key metrics:

- **Attack success rate:** Percentage of examples that successfully flipped model predictions
- **Average perturbation rate:** Percentage of words modified per example
- **Computational efficiency:** Average number of queries required per successful attack

Additionally, for successful attacks, original and perturbed texts were saved along with their corresponding labels to enable qualitative analysis of perturbation patterns. The combination of these quantitative and qualitative metrics provided comprehensive insights into the relative vulnerabilities of BERT and DistilBERT across different attack vectors, while also revealing patterns in how small text modifications can significantly impact model predictions.

6 Results

6.1 Baseline Model Performance

As shown in Table 1, both models achieved comparable baseline performance metrics on the IMDB dataset with approximately 88% accuracy prior to adversarial testing.

Table 1: BERT(Epoch1) and DistilBERT(Epoch3) Performance Metrics on IMDB Dataset

Metric	BERT	DistilBERT
Accuracy	0.888	0.89
Precision	0.889	0.89
Recall	0.888	0.89
F1 Score	0.888	0.89

Table 2: Character-level Adversarial Attack Results

Metric	BERT	DistilBERT
Dataset	IMDB	IMDB
Number of Examples	1000	1000
Successful Attacks	370	560
Failed Attacks	570	375
Skipped Attacks	60	65
Success Rate (%)	39.36	59.89

6.2 Character-level Attack Results

Table 2 presents the results of character-level attacks against both models. BERT demonstrated greater resilience to these attacks with a success rate of 39.36% (370 successful attacks out of 1000 examples), while DistilBERT proved more vulnerable with a 59.89% success rate (560 successful attacks out of 1000 examples). Both models had a similar number of skipped attacks (60 for BERT, 65 for DistilBERT).

6.3 Word-level Attack Results

Word-level attacks, as detailed in Table 3, achieved remarkably high success rates against both models. BERT showed a 99.89% success rate (939 successful attacks out of 1000 examples) with only a single failed attack, while DistilBERT exhibited a nearly identical vulnerability at 99.79% (933 successful attacks). The number of skipped attacks remained consistent at approximately 60 examples for both models.

6.4 Homoglyph Attack Results

The results for homoglyph attacks are presented in Table 4. BERT showed higher vulnerability with a 95.11% success rate (894 successful attacks), compared to DistilBERT's slightly lower 87.6% success rate (876 successful attacks). Failed attacks were more numerous for DistilBERT (64) than for BERT (46), while skipped attacks remained consistent at 60 examples for both models.

Table 3: Word-level Adversarial Attack Results

Metric	BERT	DistilBERT
Dataset	IMDB	IMDB
Number of Examples	1000	1000
Successful Attacks	939	933
Failed Attacks	1	2
Skipped Attacks	60	65
Success Rate (%)	99.89	99.79

Table 4: Homoglyph Adversarial Attack Results

Metric	BERT	DistilBERT
Dataset	IMDB	IMDB
Number of Examples	1000	1000
Successful Attacks	894	876
Failed Attacks	46	64
Skipped Attacks	60	60
Success Rate (%)	95.11	87.6

6.5 DeepWordBug Attack Results

Table 5 shows that DeepWordBug attacks achieved a 52.55% success rate (494 successful attacks) against BERT, compared to a notably higher 68.45% success rate (640 successful attacks) against DistilBERT. Failed attacks were substantially lower for DistilBERT (295) compared to BERT (446), with similar numbers of skipped examples (60-65) for both models.

7 Analysis

7.1 Comparative Model Robustness

The experimental results reveal significant differences in robustness between BERT and DistilBERT against various adversarial attacks. DistilBERT demonstrated greater vulnerability to character-level attacks (59.89% success rate) and DeepWordBug attacks (68.45% success rate) compared to BERT (39.36% and 52.55% respectively). This pattern suggests that model compression, while beneficial for efficiency, may compromise robustness against attacks that target character-level representations. Interestingly, BERT showed slightly higher vulnerability to homoglyph attacks (95.11%) compared to DistilBERT (87.6%), indicating that model size does not uniformly correlate with adversarial robustness across all attack vectors.

7.2 Attack Effectiveness Hierarchy

A clear hierarchy emerged in attack effectiveness across both models. Word-level attacks consis-

Table 5: Deepwordbug Adversarial Attack Results

Metric	BERT	DistilBERT
Dataset	IMDB	IMDB
Number of Examples	1000	1000
Successful Attacks	494	640
Failed Attacks	446	295
Skipped Attacks	60	65
Success Rate (%)	52.55	68.45

tently achieved near-perfect success rates (>99% for both models), demonstrating that semantic-preserving word substitutions represent the most formidable threat to transformer-based architectures. Homoglyph attacks followed as the second most effective methodology (>87% success for both models), leveraging the visual similarity of characters to disrupt tokenization processes. DeepWordBug and character-level attacks showed more moderate effectiveness, though still succeeded against a substantial portion of examples. This hierarchy suggests that transformer models are particularly vulnerable to attacks that preserve semantic meaning while altering the lexical form that the model processes.

7.3 Effectiveness-Efficiency Trade-offs

The relationship between attack effectiveness and computational efficiency reveals important considerations for practical security assessments. Character-level attacks required the fewest queries per successful attack (approximately 247) but demonstrated the lowest success rates, particularly against BERT. Conversely, word-level attacks demanded substantially more computational resources (approximately 473 queries per attack) but achieved the highest success rates. This inverse relationship between computational demands and success rates indicates that more sophisticated attack methods require greater resources but yield significantly higher effectiveness. From a security perspective, word-level attacks represent the optimal balance of effectiveness and computational efficiency, combining exceptionally high success rates with moderate resource requirements.

7.4 Architectural Vulnerability Implications

The consistently high effectiveness of semantically-preserving attacks across both model architectures points to fundamental vulnerabilities in how transformer models process language. The success of word-level substitutions suggests that these models, despite their contextual understanding capabilities, remain highly sensitive to lexical features. Similarly, the effectiveness of homoglyph attacks reveals critical weaknesses in tokenization processes that can be exploited without altering semantic content. These findings indicate that current adversarial vulnerabilities stem from architectural limitations rather than implementation-specific weaknesses, suggesting that more robust NLP systems will require fundamental advances in model design rather than incremental improvements to existing architectures.

8 Conclusion

This research has demonstrated that transformer-based NLP models remain highly vulnerable to adversarial text perturbations despite their impressive performance on standard benchmarks. The systematic evaluation of four distinct attack methodologies revealed that word-level and homoglyph attacks consistently achieved the highest success rates against both BERT and DistilBERT, while character-level attacks showed the greatest differentiation between model architectures. These findings highlight that current model vulnerabilities stem from fundamental aspects of transformer architecture rather than implementation-specific weaknesses, as even small text modifications can significantly compromise model predictions while preserving human readability.

Future defense strategies could include adversarial training, where models are explicitly trained on adversarial examples to improve robustness. Input preprocessing techniques could help normalize potential adversarial manipulations before they reach the model. Ensemble methods combining multiple models with different architectures could also reduce vulnerability, as attacks effective against one model may fail against others. For critical applications, human-in-the-loop systems could provide an additional layer of verification for flagged examples.

Additionally, examining the relationship between model compression techniques and adversarial vulnerability could yield valuable insights for developing more robust yet efficient models. As NLP systems become increasingly embedded in security-critical applications, addressing these vulnerabilities is essential for reliable real-world deployment.

References

[1] Steffen Eger, Gözde Gül Şahin, Andreas Rücklé, Ji-Ung Lee, Claudia Schulz, Mohsen Mesgar, Krishnkant Swarnkar, Edwin Simpson, and Iryna Gurevych. 2019. Text processing like humans do: Visually attacking and shielding NLP systems. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1634–1647, Minneapolis, Minnesota. Association for Computational Linguistics.

[2] Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 50–56.

[3] Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8018–8025.

[4] John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. TextAttack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126, Online. Association for Computational Linguistics.