Adversarial Robustness Analysis of BERT

Vamshi Sunku Mohan*[1]

[1] Center for Cybersecurity Systems and Networks, Amrita Vishwa Vidyapeetham, Amritapuri.
*corresponding author: vamshisunkumohan@gmail.com

Keywords: Interpretability, NLP, BERT.

Abstract. The research conducts an interpretability study to analyse the behavioural change in BERT while being prompted with semantically-preserved adversarial perturbations. We employ gradient-based saliency analysis to identify critical tokens. We craft minimal word substitutions that flip model predictions while preserving human-perceived meaning. Through systematic experimentation on SST-2 dataset, we show BERT's vulnerability to synonym attacks with 32% success rate while maintaining an average of 86% confidence of generating incorrect predictions. Further, we propose structural modifications to calibrate BERT in reducing confidence on producing incorrect results and reduce attack success rate for single-word flipping.

1 Introduction

Bidirectional Encoder Representations from Transformers (BERT) [1] uses an encoder-only architecture capturing the semanticity of the previous and next words while learning the context of a particular word, thus producing a bi-directional representation to model context-sensitive semantic representations of tokens. Although widely used in NLP tasks such as sentiment classification and question-answering due to these properties, their robustness in detecting adversarial inputs remains a critical concern.

This research investigates BERT's susceptibility to synonym-based adversarial attacks through a comprehensive interpretability study. We employ gradient-based saliency analysis to identify critical tokens in input text, develop minimal word substitutions to flip model predictions while maintaining human-perceived meaning. Our systematic evaluation on the Stanford Sentiment Treebank (SST-2) dataset [2] shows that attack success rates escalate from 18% with single-word substitutions to 32% with five-word substitutions and the model maintains a high confidence of about 86% while producing incorrect predictions on adversarial examples.

We further investigate the model layers and attention mechanisms to detect the components that are most vulnerable to adversarial perturbations. Additionally, we propose structural modifications to mitigate incorrect prediction confidence while maintaining good non-adversarial prediction accuracy and semanticity.

2 Proposed Approach and Results

2.1 Experimental Setup and Adversarial Prediction Results

We employ BERT-base-uncased [3] model containing 110 million parameters, 12 layers, 768 hidden dimensions on SST-2 dataset. The model is fine-tuned for 3 epochs with a learning rate of 2e-5, batch size of 16, weight decay of 0.01 and AdamW optimizer. Dataset is partitioned into 1,000 training, 500 validation and 300 test samples to enable efficient experimentation.

2.2 Adversarial Prompt Generation

To generate adversarial prompts, we first identify tokens with the most importance, replace them with their synonyms and evaluate attack success rate.

Gradient-Based Token Importance: To craft effective adversarial examples, we identify tokens that influence model predictions the most. We propose a gradient-based saliency analysis, computing the gradient of the loss with respect to input embeddings. Embeddings are calculated using "$embeddings = model.bert.embeddings.word\_embeddings(input\_ids)$" and their importance using "$importance = embeddings.grad.abs().sum(dim = -1)$".

Synonym Substitution: Tokens with high importance are substituted with the respective synonyms extracted from WordNet database [4] in NLTK library to generate perturbed sentences and are tested to check the prediction flip. BERT achieves 87.0% test accuracy on clean data, with average prediction confidence of 94.5%. However, incorrect predictions maintain 84.7%, indicating the model is miscalibrated. The attack success rate, prediction accuracy and model confidence for varying perturbations are listed in Table 1.

Number of Perturbations (words)	Attack Success Rate (%)	Prediction Accuracy (%)	Incorrect Prediction Confidence (%)
1	18.0	82.0	89.0
2	24.0	76.0	88.0
3	29.0	71.0	87.2
5	32.0	68.0	86.3

Table 1: Adversarial attack effectiveness results

We observe that attack success is 18%, 24%, 29% and 32% respectively for 1, 2, 3 and 5 words perturbed. The resultant prediction accuracy drops successively from 87% for 1-word flip to 68% for 5-word perturbation. However, we observe that model confidence in predicting errors remains considerably high with an average of 87% indicating that BERT cannot distinguish between adversarial and non-adversarial sentences.

2.3 Interpretability Analysis

We study BERT's internal mechanism using the following interpretability techniques.

Layer-Wise Vulnerability Analysis To identify transformer layers that are vulnerable to adversarial perturbations, we flatten and normalize the attention matrices and compute the attention divergence at each layer using Jensen–Shannon (JS) divergence [5] as shown in Fig. 1. JS divergence being a symmetric metric, shows the drift of adversarial text from clean input without exploding divergence under zero probability. Further, we observe that the divergence increases from 0.22 in the early layers to 0.57 in the middle layers and 0.44 in the final layers. Layers 5–10 exhibit a 159.09% higher divergence compared to the early layers, indicating significant semantic disruption occurring within the model's intermediate attention processing.

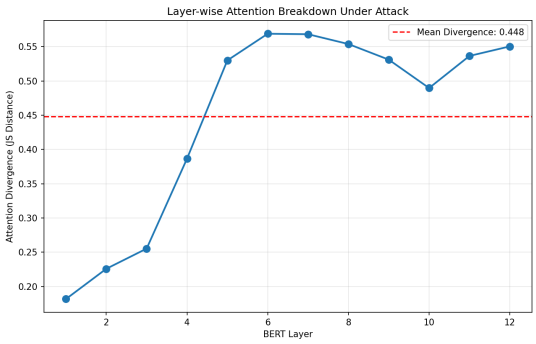

Figure 1: Layer-wise attention divergence of BERT under attack

Embedding Space Analysis We analyse the spread of feature embeddings in a 2D space using PCA. This study helps understand BERT's high confidence in predicting adversarial texts. Fig. 2a shows an overlapping distribution between adversarial and non-adversarial examples. Arrows

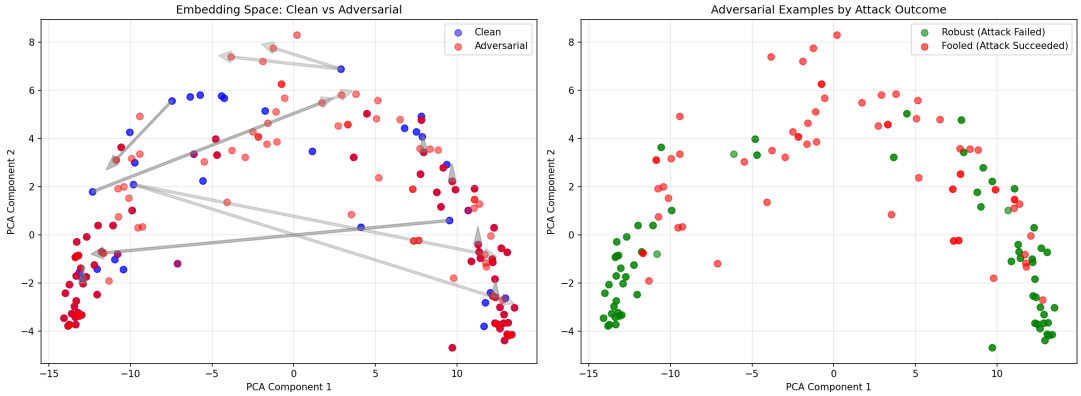

Figure 2: Embedding distribution of, (a) Clean vs Adversarial samples, (b) Succeeded vs Failed Adversarial samples

connecting each clean example to its adversarial counterpart indicate the perturbation direction and magnitude. Although some clean-adversarial pairs are distinguishable with an average L2 distance of 12.87, most overlap with no visible decision boundary, explaining the model's high error confidence.

Fig. 2b shows the PCA plot of attacks and their clean counterparts. We observe that successful attacks are spread throughout the embedding space without a clear decision boundary indicating that BERT is vulnerable to a wide range of perturbations, thus making errors unquantifiable. Further, we analyze token-level attention changes to understand the effect of perturbations on model processing. Fig. 3 shows attention redistribution between original and adversarial texts, indicating attention concentration on substituted tokens resulting in high error confidence.

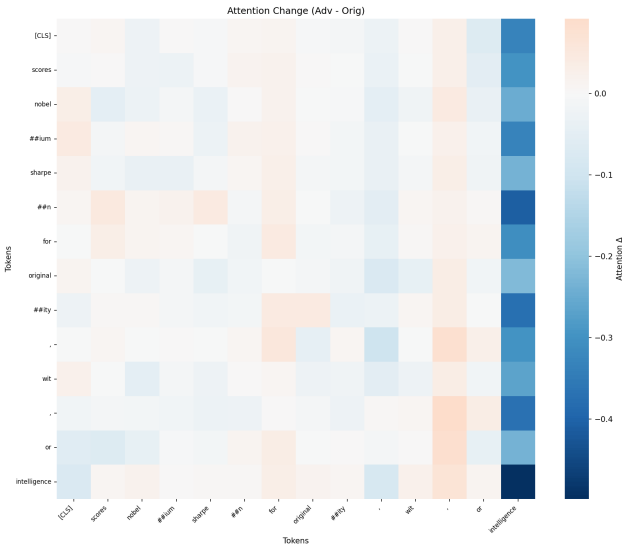

Figure 3: Token-wise change in model attention for adversarial texts

Semantic Similarity Validation Since attacks show high confidence in predicting adversarial texts, we study their semantic similarity with the clean text using SentenceTransformers package in sentence_transformers library [6] to analyse how BERT understands and distinguishes them. Fig. 4a shows that most attacks cluster around 0.9-1.0 similarity with only a few attacks in range 0.2-0.7 producing significant semantic drift. Fig. 4b reinforces Fig. 2b results showing semantically sound adversarial examples are located near to their clean counterparts. Table 2 shows that BERT has an attack success rate of 72.4% with similarity greater than 0.85. This validates semantic preservation, while the 20.7% attacks with similarity less than 0.75 shows that BERT confidently misclassifies text, indicating inability to separate original texts from adversarial texts.

3 Proposed Improvements

1. Modify activation function - Since, BERT predicts adversarial-texts with high probability, softmax activation function predicting only probability distributions without considering confidence variance may not be useful in the output layer. Hence, it could be replaced

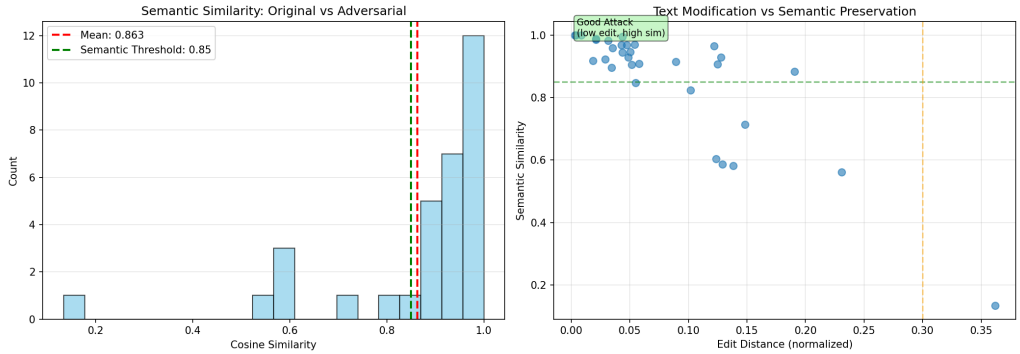

Figure 4: Semantic similarity analysis showing, (a) Number of semantically similar adversarial texts, (b) Original and adversarial texts based on L2 distance

Semanticity Range	Number of Attacks (%)	Semanticity Preservation
0.90 - 1.00	51.7	Excellent
0.85 - 0.90	20.7	Good
0.75 - 0.85	6.9	Moderate
< 0.75	20.7	Poor

Table 2: Semantic Threshold Analysis (Conducted for 30 attacks)

with Dirichlet distribution [7] that predicts upcoming words with uncertainty. This helps to reduce overconfidence on erroneous predictions.

2. Neighbor-aware token embeddings - Since single-word perturbations result in 18% success rate, single-word bidirectional embedding can be extended through contextual smoothing over multiple neighboring tokens to check the word's bidirectional context between neighbouring words and sentences. This enhancement would reduce single-token brittleness by blending each token's representation with its local context.

3. Study attention shift - Middle layers in Fig. 1 show a high attention divergence of 159.09% under attack indicating their susceptibility to attention concentration on attack tokens. To reduce the concentration, softmax function in the middle layers can be extended with Lipschitz constraints [8] and temperature scaling. This ensures that small changes in input causes minimal attention changes, thus providing robust predictions. By limiting attention shift given bounded input perturbations, this mechanism addresses the vulnerability in middle layers while maintaining standard attention in initial and final layers.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[2] Sst-2 dataset. https://huggingface.co/datasets/stanfordnlp/sst2.

[3] Bert-base-uncased. https://huggingface.co/google-bert/bert-base-uncased.

[4] Wordnet database. https://www.nltk.org/api/nltk.corpus.reader.wordnet.html.

[5] Jensen-shannon divergence. https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.jensenshannon.html.

[6] Sentencetransformers. https://huggingface.co/sentence-transformers.

[7] Dirichlet distribution. https://numpy.org/doc/2.0/reference/random/generated/numpy.random.dirichlet.html.

[8] Hongfei Xu, Qiuhui Liu, Josef van Genabith, Deyi Xiong, and Jingyi Zhang. Lipschitz constrained parameter initialization for deep transformers, 2020. https://aclanthology.org/2020.acl-main.38.pdf.