Minimizing Queries, Maximizing Impact: Adaptive Score-Based Attack and Defense for Sentiment Analysis

Yigit Efe Enhos
Brown University
yigit_efe_enhos@brown.edu

Shira Wein
Amherst College
swein@amherst.edu

Scott Alfeld
Amherst College
salfeld@amherst.edu

Abstract

While state-of-the-art large language models find high rates of success on text classification tasks such as sentiment analysis, they still exhibit vulnerabilities to adversarial examples: meticulously crafted perturbations of input data that guide models into making false predictions. These adversarial attacks are of particular concern when the systems in question are user-facing. While many attacks are able to reduce the accuracy of Transformer-based classifiers by a substantial margin, they often require a large amount of computational time and a large number of queries made to the attacked model, which makes the attacks susceptible to detection. In this work, we resolve the limitations of high query counts and necessary computational time by proposing a query-efficient word-level attack that is fast during deployment and does not compromise the attack success rate of state-of-the-art methods. Our attack constructs a dictionary of adversarial word substitutions based on prior data and leverages these substitutions to flip the sentiment classification of the text. Our attack method achieves an average of 27.49 queries—over 30% fewer than the closest competitor—while maintaining a 99.70% attack success rate. We also develop an effective defense strategy inspired by our attack approach.

1 Introduction

While large language models of recent years have made substantial progress towards accurate and fluent text generation and classification, they still exhibit vulnerabilities to adversarial examples: meticulously crafted perturbations of input data that guide models into making false predictions (Szegedy et al., 2014). This type of adversarial attack is particularly concerning for systems which may have impacts on real users, such as sentiment analysis systems that classify user engagement on social media (Wankhade et al., 2022). Work towards adversarial attack and defense methods enables both (1) investigation into vulnerabilities of existing models, and (2) explanation of model performance (Goodfellow and Jonathon Shlens, 2015; Gil et al., 2019).

While many attacks are able to reduce the accuracy of Transformer-based classifiers by a substantial margin (Gao et al., 2018; Gil et al., 2019; Ebrahimi et al., 2018; Liu et al., 2022), they are limited by the amount of computational time required and the number of queries made to the target (attacked) model (Zhan et al., 2024). These drawbacks make them susceptible to being detected by the model's owner, as single accounts making many queries often raise red flags (Maghsoudimehrabani et al., 2022). To overcome this issue, recent work has suggested query-efficient methods (Hossam et al., 2021; Berger et al., 2021; Lv et al., 2023; Wang et al., 2022; He et al., 2021), which reduce the number of queries made to the model—but still require substantial computational time during the deployment of the attack.

In this work, we resolve both limitations, that of high query counts and the computational time required, by proposing a query-efficient word-level attack. Our attack is both faster during the deployment (fewer queries and shorter attack time) than the state-of-the-art and does not compromise the attack success rate of state-of-the-art methods.

At its core, our proposed attack creates a word substitution dictionary which is gathered by first targeting the most frequent words in a prior dataset, and then scoring them based on their frequency, how much they change the model's prediction, LIME score (Ribeiro et al., 2016) and the amount they change the meaning. The scored substitutions are then leveraged for the target model, where applying the highest-scoring substitutions induces misclassification. Inspired by this property of word-level adversarial attacks, we also propose a defense method, which assumes that each sentence has been attacked and systematically reverts the highest-scoring substitutions.

As a result, our attack method achieves an average of 27.49 queries—30.05% fewer than the current state of the art (Di et al., 2020)—while maintaining a 99.70% attack success rate. Our defense method surpasses all known existing defenses in mitigating three out of four baseline attacks (Wang et al., 2021; Zhu et al., 2020; Yoo et al., 2022; Mozes et al., 2021) achieving the highest reduction in attack success rate and F1 score while maintaining the highest clean accuracy (accuracy when there is no attacked text in the testing sample). Our contributions include:

1. A novel word-level deployment-time attack against sentiment analysis systems which achieves state-of-the-art query reduction and near state-of-the-art attack success rate.
2. A novel defense method, which utilizes a dictionary of possibly harmful substitutions and reverts them, achieving state-of-the-art performance for reducing attack success rate and F1 score for popular sentiment analysis systems.
3. An open-source, ready-to-be-deployed attack benchmark suite of retrieved word substitutions for the research community.[1]

2 Related Work

Adversarial examples that deceive Transformer-based classifiers can be generated at the character-, word-, or sentence-level, or mixed across multiple levels. Our attack is primarily a word-level attack that also shares similarities to character-level attacks in computing word similarity and determining the ideal perturbations.

Word-level attacks are the most common, as they are able to generate more fluent and realistic samples compared to character-level attacks. Word-level attacks impose linguistic constraints to the search space and use linguistic thesauruses (such as WordNet (Miller, 1995) and HowNet (Dong and Dong, 2003)) to develop a larger vocabulary of synonyms. Early word-level text adversarial attacks employed various techniques, including computational graph unfolding to identify words that cause misclassification (Papernot et al., 2016), dropping or substituting key words based on part-of-speech tags (Samanta and Mehta, 2017), replacing words with GloVe-based nearest neighbors (Alzantot et al., 2018), and using swarm optimization for sememe substitution (Zang et al., 2020).

More recently, Di et al. (2020) and Ren et al. (2019) introduce TextFooler and PWWS, respectively, which greedily select words to perturb per query. TextFooler ranks word importance by measuring their impact on model confidence, while PWWS combines word saliency with classification probability to identify the most vulnerable words. While these models implement iterative synonym swaps that achieve high attack success rates, they often generate an excessive number of queries, making them detectable by systems like Stateful Query Analysis (Maghsoudimehrabani et al., 2022) and OpenAI (OpenAI, 2024), which flag and disrupt such attacks. To address the detectability of these attacks, E2A (Hossam et al., 2021) learns word saliency scores from a separate training corpus in a related domain, rather than a corpus intended for attacking. However, their method focuses only on identifying words to substitute, without ranking the effectiveness of the replacements.

In contrast, our approach goes one step further by scoring substitutions to determine the most impactful replacements. For example, when replacing *good*, our attack evaluate multiple alternatives, such as *wonderful* and *fabulous*, and determine that *good* $\rightarrow$ *wonderful* is the stronger attack. In deployment, E2A would apply both substitutions—without them being ordered in a way that prioritizes making the most impactful substitution first—potentially wasting queries on less effective perturbations. By ranking substitutions beforehand, our method ensures a more efficient attack strategy.

Adversarial defense methods typically fall into two categories: adversarial training and statistical approaches. Early works on adversarial training propose including adversarial samples in the batch (Di et al., 2020; Yoo and Qi, 2021). FreeLB (Zhu et al., 2020), which improves embedding space invariance by applying adversarial perturbations to word embeddings and minimizing the adversarial risk within a constrained region around input samples, where perturbations are optimized to maximize the loss while minimizing the adversarial vulnerability. Taking an information theoretic perspective, InfoBERT (Wang et al., 2021) leverages an Information Bottleneck regularizer and an Anchored Feature regularizer simultaneously.

Statistical approaches focus on replacing low-frequency words with more common alternatives, assuming that adversarial sentences exhibit higher

[1]https://github.com/yigit-efe-enhos/Minimizing-Queries-Maximizing-Impact-Adaptive-Score-Based-Attack-and-Defense-for-Sentiment-Analysis

perplexity than their original counterparts, including FGWS (Mozes et al., 2021). RDE (Yoo et al., 2022) analyzes the contributions of individual words to sentiment by evaluating sentiment density via gradient-based word importance.

Our simple yet effective defense strategy operates by identifying and reverting substitutions that have been empirically shown to flip sentiment labels, using a precomputed score-based substitution dictionary to recover the original inputs. Therefore, our proposed defense does not fit into either of these categories, as it does not do any adversarial training or consider sentence-level statistical probabilities. Instead, we introduce a new consideration: harmful substitutions, explicitly identifying and reversing adversarially impactful word changes to restore the model's intended predictions.

3 Attack Methodology

We consider the task where a Transformer-based text-classifier $F : \mathcal{X} \rightarrow \mathcal{Y}$, predicts a label $Y \in \mathcal{Y}$ given an input text $X \in \mathcal{X}$ where $\mathcal{X} = \{X_1, X_2, \ldots, X_N\}$ is the input space consisting of a corpus of N sentences and $\mathcal{Y} = \{Y_1, Y_2, \ldots Y_M\}$ is the output label space for M labels. The attacker's goal is to identify a sentence X_{adv} for a given sentence X such that X_{adv} conform to the following constraints:

$$F(X_{\text{adv}}) \neq F(X) \tag{1}$$

$$H(X_{\text{adv}}) = H(X) \tag{2}$$

$$\text{Sim}(X_{\text{adv}}, X) \geq \epsilon \tag{3}$$

where $H : \mathcal{X} \rightarrow \mathcal{Y}$ represents the predicted label for sentence X by a human, $\text{Sim} : \mathcal{X} \times \mathcal{X} \rightarrow [-1, 1]$ is a semantic similarity function and ϵ is the minimum similarity threshold between the original sentence and valid adversarial examples. The attacker does not know the defending model, but has query level access to it (and explicitly specifies the API of queries), where the query-level attack constitutes the threat model.

As such, the attack generates adversarial samples such that the predicted label does not match the original label of the text, but where humans would still predict the original label and semantic similarity is preserved. In addition to leading the target model to misclassification, a good attack aims to minimize the number of words perturbed and the number of queries, while maximizing semantic similarity and fluency.

In this work, we focus on identifying adversarial examples specifically for binary sentiment analysis, the task of classifying positive and negative sentiments. We thus have $\mathcal{Y} = \{1, 0\}$ as our label space where positive and negative samples are represented by 1 and 0, respectively.

3.1 Data Preparation Phase

Our proposed attack and defense require a preparation phase in which we collect data on substitutions that frequently cause model misclassification. In order to identify these substitutions, we attack sentences from the English-language IMDb dataset[2]. During the substitution gather process, we use an improved version of TextFooler (Di et al., 2020), which changes the saliency calculation L2 norm to be a signed summation instead of absolute value.

We begin by identifying the top 200 words in the target IMDb dataset that are neither stopwords nor named entities. We maintain the syntactic structure of the sentence, ensuring that substitutions are the same parts-of-speech and that punctuation is unaffected.

For each word, we select 100 reviews containing it and generate approximately 100,000 adversarial samples per sentence, recording every substitution used to create an adversarial example. Specifically, we select a word from the most frequent (non-stopword and non-entity) terms in the vocabulary, using the full IMDb movie reviews dataset, and then choose 100 reviews that include this word. Using our adapted TextFooler attack, for each of these 100 reviews, we first perform the substitution of the current word. If necessary, we then perform another substitution, and so on, until the label is successfully flipped. For each of these 100 reviews, we run attack attempts until we reach 100 successful attacks. The model tries synonym substitutions of the selected word and, if needed, other words to flip the review's label and reach the total 100,000 attacks. We record (a) the substitutions used and (b) the direction of the label change, and then use these records for the target model.

We repeat this process twice, in order to simulate real-world attack and defense conditions. First, we collect substitutions from the target model (the model we know we are attacking). Second, we create a *transfer* condition, where we collect substitutions from a similar model to the model we're attacking and then transfer those substitutions over

[2]https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews

to the target model. In Section 4.1, we analyze the performance of the final attack based on both of these methods used to gather substitutions.

The substitution-gathering stage is a one-time preprocessing step that scores candidate replacements for each token. While this scoring pass may appear costly, it is notably parallelizable: every (token, candidate) pair can be evaluated independently and distributed across CPU/GPU nodes with near-linear speedup. For reference, generating a full dictionary for IMDb ($\approx$ 25k word types $\times$ $\approx$10 candidates $\approx$ 250k scored pairs) requires under one hour on a single NVIDIA A100 and under ten minutes when shared across 8 A100 GPUs. Because this work is performed once per dataset (not per attack instance), the amortized cost per adversarial sample is negligible. At runtime, both attack and defense reduce to $O(k)$ table lookups and string replacements (with k the number of substituted tokens, typically < 5), so online overhead is comparable to a single forward pass of the classifier.

3.2 Attack Approach

The algorithm for our proposed attack for data gathering (shown in Appendix A) consists of two main steps: word saliency calculation and substitution.

After the data-gathering process is complete, our attack evaluates the gathered substitutions to construct our final attack by assigning scores that quantify their adversarial effectiveness. To systematically rank substitutions, we introduce a combined scoring function that incorporates four key components, defined mathematically in equations (5), (9), (10), and (11). The first component, change in model confidence, measures the extent to which a substitution decreases the model's confidence in the true label. Given a sentence $X = \{w_1, w_2, \ldots, w_n\}$, the model's confidence in the true label y is computed as

$$p(y|X) = \frac{1}{1 + e^{-z_y}} \tag{4}$$

where z_y is the classification logit of y. After applying a substitution $w_i \to w_i'$, the attack calculates the new confidence $p(y|X_{\text{adv}})$ for the adversarial sentence X_{adv}. The change in model confidence is then given by

$$C_{w \to w'} = p(y|X) - p(y|X_{\text{adv}}). \tag{5}$$

A larger $C_{w \to w'}$ value indicates a stronger adversarial effect.

The second component, the LIME score (Ribeiro et al., 2016), evaluates the local importance of a substitution by training an interpretable surrogate model that approximates the classifier's decision boundary. Given that LIME is not intended for word substitutions, we adapt the LIME score to better suit adversarial attacks by incorporating adversarial and non-adversarial substitutions (for more data).

Our attack generates adversarial samples $X_{\text{adv}}^{(k)}$ by substituting words in the original sentence and compute a weighted perturbation distance,

$$D(X, X_{\text{adv}}^{(k)}) = \delta_{\text{cos}}(X, X_{\text{adv}}^{(k)}) \delta_{\text{ham}}(X, X_{\text{adv}}^{(k)}) \tag{6}$$

where δ_{cos} is the cosine distance and δ_{ham} is the normalized Hamming distance. Using these distances, our attack computes local importance weights as

$$\pi_X(X_{\text{adv}}^{(k)}) = \exp\left(-\frac{D(X, X_{\text{adv}}^{(k)})}{\sigma^2}\right) \tag{7}$$

and fit a linear model to predict classifier outputs based on substituted words. Our adapted LIME formulation is given by:

$$\begin{aligned} \beta^*, \gamma^* = \operatorname*{argmin}_{\beta, \gamma} & \sum_{k=1}^{m} \sum_{j=1}^{p} \sum_{i=1}^{l_j} \Delta\pi_{j,i}^{(k)} \\ & \times \left(f(X_{\text{adv}}^{(k)}) - \left(\beta_j \cdot w_j^{\text{orig}} + \gamma_{j,i} \cdot w_{j,i}^{\text{sub}} \right) \right)^2 \\ & + \Omega(g) \end{aligned} \tag{8}$$

where β_j and $\gamma_{j,i}$ represent the contributions of original words and substitutions respectively, while $\Omega(g)$ is a regularization term preventing overfitting. The final LIME score is computed as:

$$L_{w \to w'} = |\gamma_{w,w'}| \cdot -\text{sign}(\beta_w \cdot \gamma_{w,w'}) \tag{9}$$

where $\gamma_{w,w'}$ captures the contribution of substitution $w \to w'$ and β_w represents the influence of the original word.

The third component, substitution frequency, ensures generalizability by prioritizing frequent adversarial substitutions. We define the frequency of a substitution as

$$F_{w \to w'} = \frac{N_{w \to w'}}{N_w} \tag{10}$$

where $N_{w \to w'}$ is the number of times $w \to w'$ has successfully flipped a label and N_w is the total number of times w has been substituted.

The final component, change in similarity, ensures that adversarial substitutions preserve semantic similarity while misleading the model. Using BERT-based sentence embeddings, we compute the similarity shift caused by a substitution as

$$S_{w\to w'} = \cos(e_X, e_{X_{\text{adv}}}) - \cos(e_X, e_{X_{\text{adv}}^{-w'}}) \quad (11)$$

where e_X and $e_{X_{\text{adv}}}$ are embeddings of the original and adversarial sentences, and $e_{X_{\text{adv}}^{-w'}}$ is the embedding of the adversarial sentence with w' reverted. Substitutions with higher semantic distortion receive lower scores. Our adaptive reward function is tuned via ablation studies (details in Appendix B).

After computing individual scores, we construct a combined score function to prioritize the most effective substitutions:

$$\begin{aligned}\text{Score}_{w\to w'} = C_{w\to w'} + L_{w\to w'} + \\ F_{w\to w'} - S_{w\to w'} \pm \epsilon\end{aligned} \quad (12)$$

where ϵ is an adaptive penalty or reward based on past attack success. Given that some substitutions change predictions from negative to positive, while others do the reverse, we obtain two separate substitution dictionaries: one for use when the original sentence is positive and another when it is negative. Using these scores, the attack proceeds in four phases:

1. First, the attack predicts the label of the original text to determine which dictionary to use.
2. Then, the text is segmented into sentences or smaller chunks, prioritizing segments containing high-score words.
3. Next, the highest-scoring substitutions are applied iteratively until the target model misclassifies the sentence.
4. Finally, redundant perturbations are removed to ensure minimal modification while maintaining adversarial effectiveness. Specifically, the attack reverts each substitution and checks if the model is still flipped or not.

The attack continues until the label changes or until we've tried 30 substitutions, at which point the highest-scoring combination is applied, as it has the best chance at a successful flip.

In analyzing our substitutions, we observe a lack of overlap between substitutions that lead the model to misclassify a positive sentence as negative and those that misclassify a negative sentence as positive. Specifically, the inverse of a high-scoring substitution that flips the label from 0 to 1 does not have a high score when applied in reverse (i.e., from 1 to 0). We measure this phenomenon using the Jaccard index, which calculates the overlap between two sets of substitutions. The observed overlap values are presented in Table 1, demonstrating the distinct nature of substitutions affecting different label transitions.

	$1 \to 0$	$0 \to 1$
$(1 \to 0)'$	6.35%	5.37%
$(0 \to 1)'$	5.37%	4.16%

Table 1: Overlap of Scored Substitutions and Inverses. The values represent Jaccard index percentages.

This discrepancy arises due to the frequency and contextual significance of certain words. For instance, a substitution such as *brilliant → brainy* is effective in flipping a positive label to negative, but the reverse substitution *brainy → brilliant* does not necessarily have the same effect when applied to a negative-labeled sentence. The rarity of certain words in the dataset exacerbates this mismatch, leading to distinct substitution patterns for positive and negative samples. This observation of non-overlapping substitutions ultimately inspires our defense approach, described in Section 3.4.

3.3 Attack Evaluation

We compare our attack against eight baseline black-box models, discussed in Section 2: Textbugger (Li et al., 2019), BAE (Garg and Ramakrishnan, 2020), BertAttack (Li et al., 2020), Textfooler (Di et al., 2020), SememePSO (Zang et al., 2020), Adv-OLM (Malik et al., 2021), PWWS (Ren et al., 2019), and E2A (Hossam et al., 2021). For our baselines, we follow the original paper's configurations and the target model. Textbugger is set to allow up to 50 candidate synonyms per word, while BAE has the same 50-synonym limit. BertAttack permits a maximum perturbation of 40% and up to 48 candidate synonyms. Textfooler enforces a minimum cosine similarity of 0.5 for word embeddings. SememePSO (Zang et al., 2020) is configured with a maximum of 20 iterations and a population of 60.

To evaluate these attacks comprehensively, we adopt six automatic metrics that assess effectiveness, subtlety, and efficiency. First, Attack Success Rate (ASR) measures the proportion of adversarial inputs that successfully change the model's predicted label. This is the most direct measure of attack effectiveness (Ebrahimi et al., 2018; Di et al., 2020). Second, Percentage of Perturbed Words (P%) captures the average fraction of words modified per input, where lower values indicate more efficient or "stealthy" attack (Li et al., 2020; Zang et al., 2020). Third, we use Cosine Similarity to assess the semantic similarity between the embeddings of the original and adversarial sentences, using sentence-level BERT embeddings. Fourth, we assess the change in language model Perplexity between the original and adversarial samples, reflecting the linguistic fluency degradation of adversarial texts (Garg and Ramakrishnan, 2020; Zang et al., 2020). Fifth, we count the Average Number of Queries required per successful adversarial example, where lower values indicate better query efficiency, particularly important in black-box settings (Alzantot et al., 2018; Morris et al., 2020). Finally, we use Average Attack Time per Sample to enable comparison of runtime feasibility across methods (Zang et al., 2020; Li et al., 2020).

Following Alzantot et al. (2018), we evaluate each model on a set of 1,000 examples randomly selected from the test set.

For our target model, we use a 12-layer BERT model, fine-tuned with 1000 reviews over 3 epochs using a learning rate of 2e-5.

3.4 Defense Approach

Our defense mechanism leverages the non-overlapping substitution property described in Section 3.2 to construct a robust counter-strategy (see Table 1). The defense operates by reversing adversarial substitutions when the words in the substitution dictionary are contained in the sentence.

If multiple potential reversions exist, the defense generates multiple versions of the sentence (prioritizing higher-scoring substitutions) and evaluates their predicted labels, determining the final prediction through a majority vote across these reverted sentences and the original text. In cases where conflicting reversions result in a tie, the label opposite to the original prediction is assigned, as the presence of multiple reversion paths increases the likelihood of an adversarial modification.

In addition, to counteract character-level perturbations, our defense employs a spell-checking mechanism before executing the reversion process, ensuring robustness against subtle manipulations.

3.5 Defense Evaluation

We compare our defense strategy against defense methods, including methods that utilize adversarial training and statistical approaches presented in Section 2.

First, we investigate the performance of the target model with the defense strategies implemented. Then, we utilize adversarial samples generated when evaluating the attacks by Textfooler, PWWS, BertAttack, and Textbugger to measure the reduction in attack accuracy and F1 score of the target model with the defense active.

To assess robustness, we evaluate the target models under adversarial attack using four core metrics: (1) Clean Accuracy, (2) F1 score, (3) Attack Success Rate Reduction, and (4) After Attack Accuracy. Clean Accuracy (CA) refers to the standard classification accuracy of the model on the clean (unperturbed) test set. It serves as a baseline to ensure that the defense does not degrade overall model performance (Jia and Liang, 2017; Liang et al., 2018). Given the mild class imbalance in SST-2, we report the F1 score to provide a balanced view of the model's precision and recall under adversarial conditions. Attack Success Rate Reduction (ASRR) is the reduction in attack success rate after applying a defense (Morris et al., 2020). Finally, After Attack Accuracy (AUA) represents the model's accuracy on adversarially perturbed data after an attack. It is often used to assess how resilient a defense makes the model to adversarial samples. While useful, this metric can be sensitive to the attack strategy and evaluation set design, as noted in Li et al. (2020).

4 Results & Analysis

We present the results of our attack and defense systems (Sections 4.1 and 4.2) via automatic metrics plus a human evaluation of the fluency of the attacked text (Section 4.3). Additionally, in Appendix C, we discuss how our attack and defense strategies port to non-BERT-based models.

4.1 Attack Results

The results of the automatic evaluation of our black-box attacks can be found in Table 2. Our attack and defense are naturally paired (applying and reverting the same list of word substitutions), so we intentionally ignore evaluating defense methods against our proposed attack as our proposed defense outperforms all, achieving a 99.4 F1 score. Given that the other four attacks (InfoBERT, FreeLB, RDE, FGWS) all operate on different vocabularies from which to pull synonyms, we select them to evaluate

Method	**Attack Success Rate**	**P%**	**Sim**	**Δ Perplexity**	Q_{avg}	T_{avg}
Textbugger	91.85%	7.43%	0.96	41.87	734.83	64.90
BAE	55.31%	3.01%	**0.97**	19.83	804.56	84.99
BertAttack	99.58%	4.23%	0.98	**15.40**	787.34	84.32
Textfooler	99.01%	14.06%	0.95	26.27	106.73	22.06
PWWS	97.39%	7.35%	**0.97**	20.57	1438.79	104.61
SememePSO	**100.00%**	8.13%	**0.97**	19.09	104623.62	481.95
Adv-OLM	94.22%	**3.67%**	0.96	30.46	2583.73	16.76
E2A	99.36%	4.07%	0.95	25.89	39.26	10.56
Our Attack (Base-scores)	98.51%	5.43%	0.96	21.37	29.11	8.92
Our Attack (1000 prior attacks)	99.62%	5.35%	0.94	21.17	27.70	8.80
Our Attack (2000 prior attacks)	99.70%	5.35%	0.94	21.19	**27.49**	**8.73**
Our Attack (Transfer/Base-scores)	97.73%	5.03%	0.96	19.90	31.22	9.13
Our Attack (Transfer/1000 prior attacks)	98.97%	4.99%	0.94	19.49	30.26	9.09
Our Attack (Transfer/2000 prior attacks)	99.29%	4.98%	0.93	19.42	30.01	9.06

Table 2: Performance of transfer and target model attack methods on BERT for the IMDb dataset. P% is the percentage of perturbed words, Sim is the cosine similarity between original and adversarial samples, Q_{avg} denotes the average number of queries, and T_{avg} is the average attack time per sample. In order to show how our attack improves over time, we get 1,000 and 2,000 prior attack scores for our model, by attacking the respective number of randomly chosen examples from the test set before evaluation. The best performance in each category is bolded.

the generality of our attack.

Most strikingly, we observe that our base proposed method achieves the lowest queries made to the model and the lowest average attack time, with 29.11 queries and 8.92 seconds on average, surpassing E2A (Hossam et al., 2021), the most query-efficient method in the literature. Furthermore, fewer queries are required as our scores adapt (with more prior attacks), achieving averages of 27.70 queries and 8.81 seconds and 27.49 queries and 8.73 seconds, respectively, at 1000 and 2000 prior attacks. The same increase in performance is also apparent in the transfer condition. In both the transfer and non-transfer/target model approaches, our method achieves a success rate on par with the top methods, with a lower level of perturbation and change in perplexity.

Exploring the adaptive nature further, we observe that scores with prior attack experience tend to produce better adversarial samples by every metric except semantic similarity. This is because the reward is independent of how the substitution changes the semantics and solely focuses on how successful the substitution is at flipping the label. Thus, the reward being independent of the semantics sometimes causes substitutions that scored relatively low due to the initial penalty of "Change in Similarity" score but still higher than 0 climb to be the best substitution if no other substitution is capable of changing the label consistently, i.e., in this scenario some substitutions with a high ability to change the label but still noticeably change the meaning end up still being prioritized. Therefore, though the meaning is changed, these substitutions enable strong performance by all other metrics.

4.2 Defense Results

The results of our defense approach (Table 3) suggest that our proposed defense performs the best among the baseline models in all cases (except the PWWS attack) on F1 score, ASRR and CA.

The lack of overlap between identified harmful substitutions and their inverses account for the high CA as the defense is not likely to generate adversarial samples by accident. With regard to F1 and ASRR, our defense outperforms all other defenses for three of the four attacks, and especially the TextFooler attack. This result is to be expected as our defense utilizes the same dictionary as the TextFooler attack when generating our dataset of harmful substitutions.

As mentioned in Section 4.1, our own defense naturally reverts all substitutions from the attack's substitution dictionary, achieving an F1 score of 99.4.

Finally, we assess the effectiveness of our defense in reverting adversarial substitutions in order to assess input recovery (obtaining the original version of the attacked sentence). To do so, we introduce two novel metrics: Perturbations Caught (PC) and Perturbations Reverted to Original (PRO).

PC measures the proportion of adversarial substitutions that our defense successfully identifies and reverts to any valid word. Let N_{reverted} be the number of adversarial substitutions reverted, and N_{total} the total number of substitutions introduced

Model	CA	PWWS (**WordNet**)		BertAttack (**Contextual**)		TextFooler (**Counter Fitted**)		TextBugger (**Sub-W GloVe**)	
		F1	ASRR(↓)	F1	ASRR(↓)	F1	ASRR(↓)	F1	ASRR(↓)
InfoBERT	87.3%	71.7	61.9	63.1	58.1	81.0	76.2	88.3	81.7
FreeLB	89.9%	77.3	68.4	72.4	61.5	88.4	80.5	92.4	85.3
RDE	88.0%	**88.8**	**77.8**	92.1	**89.1**	93.5	89.8	95.0	86.5
FGWS	90.2%	82.3	75.4	91.3	88.4	90.6	85.8	92.0	87.5
Our defense	**91.9%**	88.4	74.0	**92.4**	**89.1**	**97.4**	**90.5**	**95.4**	**90.0**

Table 3: Performance of defense methods on BERT for the IMDb dataset. Under each attack name (columns), we list the dictionary the attack uses to assess viable synonyms. Best statistics for each category are bolded.

by the attacker:

$$\text{PC} = \frac{N_{\text{reverted}}}{N_{\text{total}}} \times 100 \tag{13}$$

PRO quantifies how often the defense restores perturbed words to their exact original form. Let N_{original} denote the number of substitutions exactly reverted to the original token:

$$\text{PRO} = \frac{N_{\text{original}}}{N_{\text{reverted}}} \times 100 \tag{14}$$

PC measures the proportion of adversarial substitutions that our defense successfully identifies and replaces with any valid word. A higher PC suggests that the defense can often identify perturbations, though it does not necessarily guarantee a perfect restoration to the original input. Meanwhile, PRO quantifies how often the defense restores perturbed words to their exact original form, reflecting its precision in reconstructing the input. We compare our approach with the FGWS defense method, the only prior defense capable of input recovery, despite the concept not being mentioned in the paper.

Our method exhibits a lower amount of perturbations caught (PC of 22.1%) compared to FGWS (68.9%). While our PC is lower than FGWS (which is not preferable), when our defense does revert a word that is perturbed, it is much more likely to recover the exact original word, achieving a PRO of 78.7%, far surpassing FGWS's 42.0%. Empirically, we observe that while our defense makes unnecessary reversions via lower PC, it excels at correctly undoing adversarial modifications to their original versions—something FGWS struggles to do. This positions our method as a meaningful step forward in input recovery for sentiment analysis, demonstrating the potential of adversarial defenses not just to infer sentiment correctly but to reconstruct the original sentence.

4.3 Human Evaluation for the Attack

In order to evaluate the performance of our attack and ensure that the perturbed text maintains the original sentiment (i.e., the substitution only tricks the model, and would not trick a human) and that the semantics of the text is maintained after the attack, we conduct a human evaluation study with five annotators. The annotators are native English speakers who teach English. The survey consists of two parts: a human binary classification of attacked texts, and a human semantic similarity evaluation of the attacked text and the original text.

In the first part, participants are presented with 100 randomly chosen adversarially perturbed sentences from our dataset, and asked to classify them as either positive or negative. Notably, annotators overwhelmingly align their sentiment labels with the original, with the number of sentences labeled according to the original sentiment ranging from a low of 91 to a high of 98 across the five annotators, and an average of 95.6 sentences classified as the original label.

To quantify agreement among annotators, we compute Cohen's kappa and obtain a value of 0.92, indicating extremely high agreement (Cohen, 1960). This high agreement reflects the annotators' strong tendency to classify adversarially perturbed sentences with their original sentiment. In total, 22 sentences are classified with their adversarial label (out of 500 annotations), which can be attributed to the fact that each participant misclassified different sentences uniquely.

The second part of the survey focuses on evaluating the similarity between the original and adversarially perturbed sentences. We ask annotators to rate the similarity of each adversarial sentence to its original counterpart on a 5-point scale, with 1 indicating "completely different" and 5 indicating "nearly identical." Overall, the average similarity score across all annotators is 3.6. Per annotator, similarity scores ranged from a low of 2.7 to a high of 4.0. In addition, annotators provided qualitative feedback, frequently describing the adversarial sentences as "clunky," "unnatural," or "grammatically awkward."

5 Conclusion

In this paper, we propose an adversarial attack method for sentiment analysis systems, which is based on the notion of identifying harmful substitutions. Our two-stage attack begins with a data-gathering process where all substitutions previously involved in successful adversarial samples are recorded and then scored via four scoring functions. This approach achieves the lowest query count and fastest execution in the literature while maintaining an on-par attack success rate.

An observation that adversarial attacks tend to use a consistent set of word substitutions then inspires our defense model, which then reverses these possibly harmful substitutions. This defense outperforms all other novel approaches in terms of F1, ASRR, and CA against three of the four baseline attacks and shows substantial progress in input recovery. In addition, our defense offers a novel outlook on ways to prevent text adversaries beyond statistical and adversarial training approaches.

Limitations

One key challenge is that the most effective substitutions vary across models: BERT, WordCNN, and WordRNN each have different highest-scoring substitutions for the same word. This variation reduces the effectiveness of transfer-based defenses, as substitutions gathered from one model may not be optimal for another (see Appendix C).

In addition, our method requires an early data preparation phase, where substitutions are gathered and scored, which demands both time and computational resources. This preprocessing step, while improving attack efficiency during deployment, introduces overhead that may not be ideal for real-time adversarial scenarios.

We develop and evaluate both our attack and defense on English data, so it is unclear how these approaches might perform on other languages. Multilingual and non-English systems introduce morphology and tokenization variability (e.g., rich inflection, compound splitting) that reduce the transferability of English-centric substitutions and may cause the defense to miss (or incorrectly revert) adversarial changes, especially in code-mixed posts.

Finally, while our defense and attack show comparable strength to state-of-the-art methods, even being superior in some aspects, they lack generality. Future work can pursue ideas presented in this paper and apply them to the broad text classification task, in particular non-binary classification tasks. Although we focus on binary sentiment analysis of movie reviews, the attack/defense paradigm we introduce–computing a dictionary of high-impact substitutions and then applying/reverting them at deployment–naturally extends to other text classification tasks that admit token-level explanations (e.g., hate-speech detection, topic labeling, or aspect-based sentiment). In such cases, the same scoring pipeline can be reused by swapping the task-specific classifier and recalibrating similarity and fluency constraints to the target domain. That said, two caveats limit external validity. First, distribution shift in surface form is severe on social media platforms; code-switching, slang, emojis, hashtags, elongated words, and creative orthography can degrade both our substitution scores and our defense's reversion accuracy. Attackers could exploit this by computing platform-specific substitution dictionaries in advance, and then executing queries that evade rate limits and stateful detectors.

Ethical Considerations

Any attack method, including ours, is susceptible to abuse. In this case, our method could be exploited for sentiment analysis systems on social media platforms or other user-facing systems. We mitigate this risk by introducing a highly effective defense system as well, but that does not negate the possibility of the attack's abuse.

Acknowledgments

We thank anonymous reviewers and the annotators for our human evaluation.

References

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896, Brussels, Belgium. Association for Computational Linguistics.

Nathaniel Berger, Stefan Riezler, Sebastian Ebert, and Artem Sokolov. 2021. Don't search for a search method — simple heuristics suffice for adversarial text attacks. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8216–8224, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

Jin Di, Joey Tianyi Zhou Zhijing Jin, and Peter Szolovits. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. *Association for the Advancement of Artificial Intelligence*.

Zhendong Dong and Qiang Dong. 2003. Hownet - a hybrid language and knowledge resource. In *International Conference on Natural Language Processing and Knowledge Engineering, 2003. Proceedings. 2003*, pages 820–824.

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.

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.

Siddhant Garg and Goutham Ramakrishnan. 2020. BAE: BERT-based adversarial examples for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6174–6181, Online. Association for Computational Linguistics.

Yotam Gil, Yoav Chai, Or Gorodissky, and Jonathan Berant. 2019. White-to-black: Efficient distillation of black-box adversarial attacks. 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 1373–1379, Minneapolis, Minnesota. Association for Computational Linguistics.

Ian J. Goodfellow and Christian Szegedy Jonathon Shlens. 2015. Explaining and harnessing adversarial examples. *International Conference on Learning Representations*.

Xuanli He, Lingjuan Lyu, Qiongkai Xu, and Lichao Sun. 2021. Model extraction and adversarial transferability, your BERT is vulnerable! In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2006–2012, Online. Association for Computational Linguistics.

Mahmoud Hossam, Trung Le, He Zhao, and Dinh Phung. 2021. Explain2attack: Text adversarial attacks via cross-domain interpretability. In *2020 25th international conference on pattern recognition (ICPR)*, pages 8922–8928. IEEE.

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.

Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2019. Textbugger: Generating adversarial text against real-world applications. In *Proceedings 2019 Network and Distributed System Security Symposium*, NDSS 2019. Internet Society.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. BERT-ATTACK: Adversarial attack against BERT using BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6193–6202, Online. Association for Computational Linguistics.

Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. 2018. Deep text classification can be fooled. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4208–4215. International Joint Conferences on Artificial Intelligence Organization.

Aiwei Liu, Honghai Yu, Xuming Hu, Shu'ang Li, Li Lin, Fukun Ma, Yawen Yang, and Lijie Wen. 2022. Character-level white-box adversarial attacks against transformers via attachable subwords substitution. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7664–7676, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Wenjie Lv, Zhen Wang, Yitao Zheng, Zhehua Zhong, Qi Xuan, and Tianyi Chen. 2023. Buffersearch: Generating black-box adversarial texts with lower queries. *Preprint*, arXiv:2310.09652.

Mohamed Maghsoudimehrabani, Ali Dehghantanha Amin Azmoodeh, and Behrouz Zolfaghari. 2022. Proactive detection of query-based adversarial scenarios in nlp systems. *CCS '22: 2022 ACM SIGSAC Conference on Computer and Communications Security*.

Vijit Malik, Ashwani Bhat, and Ashutosh Modi. 2021. Adv-OLM: Generating textual adversaries via OLM. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 841–849, Online. Association for Computational Linguistics.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

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.

Maximilian Mozes, Pontus Stenetorp, Bennett Kleinberg, and Lewis Griffin. 2021. Frequency-guided word substitutions for detecting textual adversarial examples. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 171–186, Online. Association for Computational Linguistics.

OpenAI. 2024. Openai api documentation.

Nicolas Papernot, Ananthram Swami Patrick McDaniel, and Richard Harang. 2016. Crafting adversarial input sequences for recurrent neural networks. *MILCOM 2016-2016 IEEE Military Communications Conference, pages 49–54. IEEE.*

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1085–1097, Florence, Italy. Association for Computational Linguistics.

Marco Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 97–101, San Diego, California. Association for Computational Linguistics.

Suranjana Samanta and Sameep Mehta. 2017. Towards crafting text adversarial samples. *Preprint*, arXiv:1707.02812.

Christian Szegedy, Ilya Sutskever Wojciech Zaremba, Dumitru Erhan Joan Bruna, and Rob Fergus Ian J. Goodfellow. 2014. Intriguing properties of neural networks. *International Conference on Learning Representations*.

Boxin Wang, Shuohang Wang, Yu Cheng, Zhe Gan, Ruoxi Jia, Bo Li, and Jingjing Liu. 2021. Infobert: Improving robustness of language models from an information theoretic perspective. In *International Conference on Learning Representations*.

Zhen Wang, Yitao Zheng, Hai Zhu, Chang Yang, and Tianyi Chen. 2022. Transferable adversarial examples can efficiently fool topic models. *Computers and Security*, 118:102749.

Mayur Wankhade, Annavarapu Chandra Sekhara Rao, and Chaitanya Kulkarni. 2022. A survey on sentiment analysis methods, applications, and challenges. *Artificial Intelligence Review*, 55(7):5731–5780.

Jin Yong Yoo and Yanjun Qi. 2021. Towards improving adversarial training of NLP models. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 945–956, Punta Cana, Dominican Republic. Association for Computational Linguistics.

KiYoon Yoo, Jangho Kim, Jiho Jang, and Nojun Kwak. 2022. Detection of adversarial examples in text classification: Benchmark and baseline via robust density estimation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3656–3672, Dublin, Ireland. Association for Computational Linguistics.

Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun. 2020. Word-level textual adversarial attacking as combinatorial optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6066–6080, Online. Association for Computational Linguistics.

Pengwei Zhan, Jing Yang, He Wang, Chao Zheng, and Liming Wang. 2024. Rethinking word-level adversarial attack: The trade-off between efficiency, effectiveness, and imperceptibility. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 14037–14052, Torino, Italia. ELRA and ICCL.

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. 2020. Freelb: Enhanced adversarial training for natural language understanding. In *International Conference on Learning Representations*.

A Algorithm for Generation of Adversarial Samples

B Ablation Studies for Reward Function

Our adaptive reward function maintains a score for each substitution candidate, which is updated dynamically as the attack progresses. Substitutions that are frequently attempted but rarely result in successful adversarial examples have their scores reduced, while those that consistently lead to model misclassifications see their scores increase. This empirical reinforcement mechanism helps prioritize substitutions that are effective in practice, allowing the attack to converge on impactful word swaps over time.

To prevent over-exploration of weak candidates, we introduce a hyperparameter ϵ that acts as a cutoff: substitutions whose scores fall below ϵ are excluded from future consideration. We conduct an ablation study to understand the effect of ϵ on Attack Success Rate (ASR), testing values in the range $\{0, 0.1, 0.3, 0.5, 0.7, 1, 1.2\}$, while holding $\psi = 50$ constant, which denotes the average number of times a substitution is attempted across attacks. As shown in Figure 1, ASR remains stable between $\epsilon = 0.1$ and $\epsilon = 0.5$, with the best performance observed at $\epsilon = 0.1$.

Algorithm 1 Generation of Adversarial Samples

```
1:  **Input:** Sentence example $X = \{w_1, w_2, \dots, w_m\}$ from
    a review, target model $F$, maximum number of
    adversarial words per sentence $n$, vocabulary of stop
    words from NLTK $\text{Vocab}_{\text{StopWords}}$
2:  **Output:** Set of adversarial words
    $\mathcal{X}_{\text{adv}} = \{X_{\text{adv}_1}, X_{\text{adv}_2}, \dots, X_{\text{adv}_k}\}$ for $k \leq n$

3:  Initialize sets $\mathcal{X}_{\text{adv}}$, $\text{Sub}_{\text{Words}}$, and dictionary $\mathcal{S}$

4:  while $|\mathcal{X}_{\text{adv}}| < n$ do
5:      while $|\text{Sub}_{\text{Words}}| < m$ do
6:          $X \leftarrow X.\text{replace}(\text{word in Sub}_{\text{Words}}, [\text{MASK}])$
7:          for each word $w_i$ in sentence $X$ do
8:              if $w_i \notin \text{Vocab}_{\text{StopWords}} \cup \text{Sub}_{\text{Words}}$ then
9:                  Calculate saliency $S_i$
10:                 $\mathcal{S}[w_i] \leftarrow S_i$
11:             end if
12:         end for
13:         $w_{\text{min}_s} \leftarrow \arg\min_{w_i} \mathcal{S}[w_i]$
14:         $\text{Sub}_{\text{Words}} \leftarrow w_{\text{min}_s}$
15:         Generate a set of substitutions SUB
16:         for each substitution $s \in$ SUB do
17:             $X_{\text{adv}} \leftarrow X.\text{replace}(w_{\text{min}_s}, s)$
18:             if $F(X_{\text{adv}}) \neq F(X)$ then
19:                 $\mathcal{X}_{\text{adv}} \leftarrow \mathcal{X}_{\text{adv}} \cup \{X_{\text{adv}}\}$
20:                 if $|\mathcal{X}_{\text{adv}}| \geq n$ then
21:                     return $\mathcal{X}_{\text{adv}}$
22:                 end if
23:             end if
24:         end for
25:     end while
26: end while
27: return $\mathcal{X}_{\text{adv}}$
```

We do not conduct further experiments with ψ, as fixing it allows us to isolate the influence of ϵ. However, we note that the ideal choice of ϵ may depend on the value of ψ, as both affect the score convergence speed of the adaptive system. A full grid search over both parameters is left to future work.

C Remarks on Non-BERT Models

To evaluate the transfer capabilities of both the attack and defense, we run the same experiments on a WordCNN and WordRNN fine-tuned with 1,000 movie reviews from the IMDb dataset over 10 epochs, using substitution words gathered from our original BERT model. The results of the attack are displayed in Table 4. We observe that our attack performs similarly on both models, achieving a 99.07% ASR for the WordRNN and a 98.47% ASR for the WordCNN while maintaining a low number of queries, with 29.08 and 28.68 average queries, respectively.

Based on these two metrics, compared to BERT, the attack performs slightly worse on WordCNN and WordRNN when substitutions are gathered from the original target model. However, it performs better in terms of query efficiency when substitutions are gathered from a transfer model. This improvement in query efficiency does not necessarily indicate a stronger attack on WordCNN and WordRNN, as these models are inherently easier to fool, with adversarial attacks generally achieving higher ASR and lower Q_{avg} on these models (Ebrahimi et al., 2018).

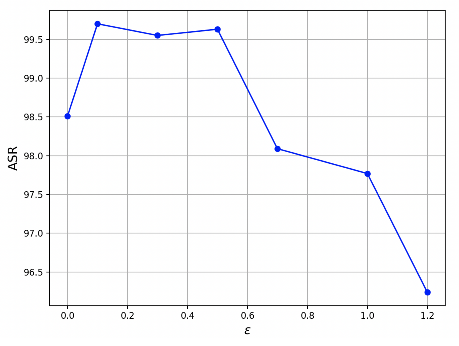

Figure 1: Impact of ϵ on ASR over 2000 adversarial samples.

While applying the substitutions gathered from BERT is successful as an attack strategy, it is ineffective when deployed as a defense for WordCNN and WordRNN (Table 5). Empirically, we attribute this to the fact that words assigned the highest scores when gathered from BERT do not necessarily retain the same high scores when gathered from WordCNN or WordRNN. Consequently, our defense strategy, which prioritizes reverting the highest-scoring adversarial substitutions, replaces words with alternatives that do not have the greatest impact in flipping the label, thereby reducing its effectiveness.

We demonstrate this in two ways: first, by gathering substitution data directly from WordCNN and WordRNN and using them for defense, and second, by computing word saliency (i.e., the impact of individual words on the predicted label) for 100 sentences across each of the three models. In Table 5, we observe that when words are gathered from a specifically tailored model (whether WordCNN or WordRNN), the defense performs similarly to our original results. This confirms that the defense strategy—ranging from scoring functions to deployment—is fundamentally sound and that the observed shortcomings arose from an improper data-gathering process.

These results indicate that prior knowledge of the structure of the model being attacked gives a

Model	WordCNN				WordRNN			
Evaluation Metric	**ASR**	**Q$_{avg}$**	**P%**	**Sim**	**ASR**	**Q$_{avg}$**	**P%**	**Sim**
Transfer (from BERT)	96.77%	35.26	5.20%	0.96	96.40%	33.64	5.89%	0.95
Transfer w/ 1000 prior attacks	98.02%	30.65	5.12%	0.92	98.92%	29.95	5.82%	0.95
Transfer w/ 2000 prior attacks	98.47%	28.68	5.01%	0.92	99.07%	29.08	5.11%	0.93

Table 4: Attack performance on WordCNN and WordRNN architectures with data gathered from a BERT model.

Source of Adversarial Words	Defended Model	Clean Accuracy (CA)	PWWS F1	TextFooler F1	TextBugger F1
BERT	WordCNN	62.34%	70.01	73.66	79.99
BERT	WordRNN	65.01%	73.94	78.57	75.40
WordCNN	WordCNN	90.46%	86.12	95.51	**93.09**
WordCNN	WordRNN	77.16%	83.79	85.49	88.40
WordRNN	WordCNN	72.47%	75.00	78.63	89.12
WordRNN	WordRNN	**90.96%**	**88.28**	**97.51**	92.59

Table 5: Performance of our defense method when trained on adversarial word substitutions gathered from different attacking models. The **Source of Adversarial Words** column indicates which model is attacked to generate adversarial substitutions used during training. The **Defended Model** column shows which model is being evaluated under defense. We report Clean Accuracy (CA) and F1 scores against three attack methods (PWWS, TextFooler, and TextBugger).

substantial advantage to the attacker.

D Remarks on Naturalness

One possible intuition is that an attack in the field succeeds primarily because the substituted sentences become “unnatural,” and therefore the model fails on text it would never encounter in a realistic setting. To examine this claim, we analyze 500 adversarial sentences and compute (1) bigram-based perplexity as a proxy for surface fluency (where higher perplexity indicates less natural text), (2) semantic similarity to the original sentence, and (3) the model’s confidence in the *true* label. Figure 2 plots these three variables jointly.

As the plot shows, the classifier does not simply fail when the text becomes less natural: a large cluster of high-perplexity sentences (highly unlikely under a bigram language model) are still classified correctly, while several low-perplexity, human-like sentences succeed in flipping the prediction. This confirms that the attack is not merely exploiting a “fluency weakness.”

This also means that defenses which rely only on fluency filtering (e.g., perplexity thresholds, grammar detectors) would block only a subset of attacks. Effective defenses must model task-specific semantics rather than assume that “unnatural $\rightarrow$ adversarial” holds in general.

Overall we note that naturalness degradation is a side effect, not the cause, of model failure. The model can correctly classify many unnatural sentences, so robustness cannot be reduced to surface fluency alone.

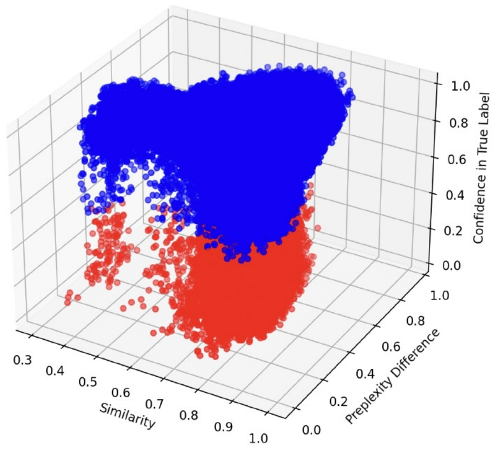

Figure 2: Scatter plot of 500 adversarial samples showing bigram-based perplexity (x), semantic similarity (y), and model confidence in the true label (z). Points in red represent *successful* adversarial attacks (label flipped), while blue points represent *unsuccessful* attacks (label preserved). Many high-perplexity (i.e., syntactically degraded) sentences remain correctly classified, while several fluent sentences still flip the label, indicating that lack of naturalness is neither necessary nor sufficient for attack success.