Proceedings of the 2nd International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/96/20241306

A Review of Adversarial Attacks and Defense Techniques in Text Processing Models

Ruihan Wang

Hainan International College, Communication University of China, Hainan, China

202329013071n@mails.cuc.edu.cn

Abstract. With the rise of neural networks, the need for accuracy, robustness, and security has increased. Research has shown that small, carefully crafted perturbations, known as adversarial examples, can deceive models and lead to incorrect predictions. Current research focuses on the image domain, while there is a notable lack of exploration in the text domain, due to its discrete nature. This paper reviews adversarial attack techniques and defense strategies in text-based neural network models, aiming to improve the security and resilience of these models in practical applications. Adversarial examples, which can deceive models with small perturbations, expose vulnerabilities in their robustness and security. Techniques such as TextFooler focus on synonym replacement for generating adversarial examples, while Text Random Smooth (Text-RS) enhances defense through adaptive noise strategies. The research of search space aims to explore the feature of that, proposing search space for Imperceptibility (SSIP) and Search Space for Effectiveness (SSET) to estimate the different attack methods. Furthermore, the Chinese Variation Graph Integration (CHANGE) method improves the resilience of Chinese language models by leveraging variation graphs. These advancements highlight the importance of developing effective generation and defense mechanisms for adversarial examples in text processing models. Future research should enhance adversarial example techniques, explore efficient defense strategies, and investigate transferability to improve the security and robustness of text processing models.

Keywords: Adversarial examples, Text adversarial attacks, Neural networks robustness, Large language mode.

1. Introduction

With the advancement of neural network technology and its applications across various fields, the demands for its accuracy, robustness, and security have progressively increased. Early studies revealed that although neural networks perform well on standard test datasets, they tend to produce significantly incorrect predictions when confronted with carefully crafted small perturbations, known as adversarial examples [1]. These adversarial examples appear almost identical to the original inputs from a human perspective, but they deceive the models, causing them to make erroneous predictions.

Adversarial attack research initially focused on computer vision, particularly in generating adversarial perturbations for image classification tasks. For instance, the Fast Gradient Sign Method (FGSM) was proposed by Goodfellow et al. to create adversarial examples by introducing small pixel-level changes to the original samples [2]. Building on this, researchers developed more complex attack methods, such as Projected Gradient Descent [3] and the Carlini & Wagner attack [4], which refined and strengthened the effectiveness of adversarial attacks. As adversarial attack research deepened, attention shifted to adversarial attacks in the field of Natural Language Processing (NLP). In contrast to the continuous nature of images, the discrete nature of text data presents unique challenges in generating adversarial examples. Minor textual changes, such as character substitution or synonym replacement, can alter semantic meaning. Researchers have developed various methods for adversarial attacks in NLP. Huang et al. focus on creating adversarial examples in NLP that preserve the semantics of the original text [5], while other research explores the generation of adversarial examples in NLP by targeting specific words and guiding sentence generation [6].

The research on adversarial examples in the field of text is still relatively underdeveloped. Although significant progress has been made in generating and defending against adversarial examples in areas like computer vision, the text domain presents unique challenges. In this work, a review on recent advancements in adversarial examples in the text domain will be presented. TextFooler is introduced as a black-box attack technique focusing on word importance ranking and substitution with semantically similar words [7]. Search space considerations in adversarial attacks are discussed, highlighting trade-offs between efficiency, effectiveness, and imperceptibility [8]. Text-RS explores noise injection and smoothing for improved defense [9], while CHANGE focuses on enhancing the robustness of Chinese language models against adversarial attacks using pinyin, visual, and character variations [10]. These methods offer insights into the generation and defense of adversarial examples in text models.

This review contributes to the field by consolidating the latest techniques and findings in adversarial example generation and defense for text-based models, providing valuable insights into the current challenges and future directions for research in this rapidly evolving field.

2. Overview of Adversarial Example Techniques

2.1. Attack Techniques

Based on the amount of information required to generate adversarial examples, attack techniques can be classified into three categories:

White-box attacks: In white-box attacks, the attacker has full knowledge of the target model's internal information, including its parameters, architecture, and gradient information. Using this information, attackers can directly compute the direction of input perturbations to generate strong adversarial examples.

Black-box attacks: Black-box attacks do not require prior knowledge of the model's internal details. Instead, attackers infer the model's behavior through input-output interactions. In black-box attacks, optimization or query-based methods are used to generate adversarial examples, such as evolutionary algorithms or surrogate models trained to mimic the target model.

Gray-box attacks: Gray-box attacks combine white-box and black-box methods. In some stages of generating adversarial examples, white-box methods are used, while black-box methods are employed in other stages. This approach allows effective generation of adversarial examples even with limited information.

2.2. Defense Techniques

Defense techniques aim to improve the model's ability to withstand adversarial attacks. These techniques can be broadly categorized into two types:

Adversarial example detection: This technique detects adversarial examples within the input data, distinguishing them from normal inputs for separate handling. Detection methods often rely on feature engineering, statistical methods, or inconsistencies in the model's behavior to identify adversarial examples.

Adversarial training: Adversarial training is a common defense method that improves a model's robustness by incorporating adversarial examples into the training process. This approach helps the model learn to defend against similar attacks in the future. However, adversarial training requires sufficiently strong adversarial examples during training, and even with this technique, models may still be vulnerable to newly crafted adversarial examples.

3. Current State of Adversarial Example Techniques

3.1. TextFooler

TextFooler serves as a robust baseline for natural language attacks, particularly in black-box settings [7]. It can quickly generate adversarial examples, compelling target models to produce incorrect predictions. Figure 1 showcases the main steps.

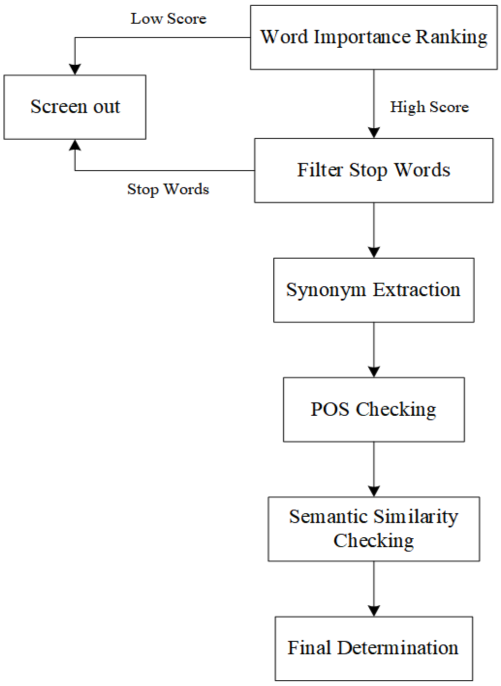

Figure 1. Main step of Textfooler

Step 1: Word Importance Ranking

For the sentence of n words $X = \{w_1, w_2, \ldots, w_n\}$, I_{w_i} is the score to measure the influence that the word w_i has on the classification result F (X) = Y. The calculation function is defined as follows,

$$I_{w_i}\begin{cases} F_Y(x) - F_Y(X_{\backslash w_i}), if\ F(X) = F(X_{\backslash w_i}) = Y \\ F_Y(x) - F_Y(X_{\backslash w_i}) + (F_{\overline{Y}}(x) - F_{\overline{Y}}(X_{\backslash w_i})), \\ if\ F(X) \neq F(X_{\backslash w_i}) \end{cases} \quad (1)$$

The sentence after deleting the word w_i is expressed $X_{\backslash w_i} = \{w_1, w_2, \ldots, w_{i-1}, w_{i+1}, w_n\}$. FY(·) is used to represent the prediction score for the Y label.

After ranking the words by I_{w_i}, filter out stop words derived from NLTK[2] and spaCy[3].

Step 2: Word Transformer

This step is to find semantically similar words to replace high-ranked ones in step1. CANDIDATES is a set of all possible replacements, including the synonyms of I_{w_i}. The standard of choosing CANDIDATES is cosine similarity. Empirically, set the largest cosine similarity δ to be 0.7 and the top synonyms N to be 50.

There are two other confirmation steps, Part-of-speech (POS) checking and semantic similarity checking. POS is to ensure the correctness of grammar.

If there are existing candidates capable of altering the target model's prediction, the word with the highest semantic similarity score is selected from these successful candidates. If not, select the word with the least confidence in label y as the best replacement word w_i, repeating Step 2 to convert the next selected word.

In terms of attack efficacy, TextFooler surpasses prior methodologies regarding both success rate and perturbation rate, while simultaneously preserving semantic integrity, grammatical correctness, and alignment with human classification. Moreover, this approach demonstrates linear computational complexity relative to text length, indicating a high degree of efficiency. Specifically for the BERT model, in classification tasks, TextFooler can diminish prediction accuracy to approximately 5%-7%, whereas in text entailment tasks, it can reduce accuracy by around 9%-22%.

3.2. Search Space

The essay emphasizes the impact of search spaces, focusing on the three key characteristics of that [8], efficiency, effectiveness, and imperceptibility.

The section on ablation studies aims to systematically remove or alter different components within the search space to assess their impact on the efficiency, effectiveness, and imperceptibility of word-level adversarial attacks. Through this series of experiments, researchers seek to understand how various search space configurations influence the performance of adversarial attacks. Based on that, it puts forward the conclusion of the impact of the three factors.

First, constraints aimed at improving one aspect can negatively affect others. Second, constraints targeting imperceptibility may worsen other imperceptibility metrics, requiring a comprehensive assessment of all related factors. Third, attack efficiency is highly sensitive to search space changes, often more so than effectiveness and imperceptibility. Finally, balancing efficiency, effectiveness, and imperceptibility is challenging, as improvements in one area can hinder others. Previous research often lacks detailed search space settings and comprehensive evaluations, leading to unfair comparisons and insufficient assessments.

Depending on the results, the paper proposed Search Space for Imperceptibility (SSIP) and Search Space for Effectiveness (SSET), aiming to estimate the different attack methods in specific standardized search spaces. Table 1 showcases the principles of data extraction.

Table 1. The feature of SSIP and SSET

SSIP	SSET
Emphasizing the grammatical correctness and semantic consistency of adversarial examples.	Emphasizing the success rate of attacks, i.e., the ability to induce incorrect outputs from the model.
Prioritizing lexical changes that have a significant impact on the model’s output but are less noticeable to human readers.	Allowing more lexical changes as long as they effectively alter the model’s predictions.
Suitable for scenarios requiring high-quality adversarial examples, such as evaluating model robustness or augmenting training data.	Suitable for scenarios requiring a large number of effective adversarial examples, such as assessing model robustness or increasing the diversity of training data.

Under different search Spaces, SSIP and SSET re-evaluated Genetic Algorithm (GA), TextBugger, TextFooler and other methods. For GA method, on AG News data set, the attack success rate under SSIP increased from 14.84% to 37.19%, while Sample Average (S.A.) reduced from 2965 to 1798. The result indicates that the SSIP and SSET have improved in different dimensions, achieving a commendable balance among efficiency, effectiveness, and stealthiness.

3.3. Text Random Smooth

The essay describes a process of generating noise and applying smoothing methods to handle it [9]. It initially showcases the steps involved in generating noise, including 3 main parts.

Given a text $x \in X$ and its corresponding word embeddings $x_e \in X_e$. To simulate the perturbation, inject a range noise, labeled as ξ. f is defined as the targeted function.

Perturbation loss: The first part of noise is perturbation loss to estimate the consistent of on noisy and noise-free texts:

$$L_s = \left\|f_p(f_e(x)) - f_p(f_e(x) + \xi)\right\|_2 \tag{2}$$

Triplet loss: The next part involves triplet loss, which aims to reduce the discrepancy between the embedding values of synonyms while simultaneously enhancing the differentiation among other words, described as function 3:

$$L_{tr} = \frac{1}{k}\sum_{w' \in Syn(w,k)}\left\|f_e(w) - f_e(w')\right\|_2 - \frac{1}{m}\sum_{\widehat{w} \notin Syn(w,k)}\left\|f_e(w) - f_e(w')\right\|_2 \tag{3}$$

Adaptive variable: The noise ξ is modeled as Gaussian noise $N(0, \sigma^2)$, where σ denotes the maximum Euclidean distance among the top-k synonyms. However, when k is large, the words within the synonym set may exhibit greater semantic differences and larger Euclidean distances, leading to increased noise levels that could reduce the model's robustness. Therefore, an adaptive variable is introduced to adjust the amount of noise injected into the word embeddings.

The noise is defined as $\xi \sim \mathrm{N}\left(0, \mathrm{diag}\left(\left\{\mathrm{a_i}\sigma_\mathrm{i}^2\mathrm{I}\right\}_{\mathrm{i}=1}^{\mathrm{n}}\right)\right)$, initialized to 1. It will be optimized alongside the model parameters during training.

By adding the generally used classification loss $\mathrm{L_{cls}}$, three types of loss are integrated. The overall training objective is as follows:

$$L(x, y) = L_{cls} + \mu_1 L_s + \mu_2 L_{tr} \tag{4}$$

As previously mentioned, training with continuous perturbations, when incorporated into the defense model's training process, can lead to a broader optimization space and improved training efficiency. The following Theorem examines the addition of noise to word embeddings to ensure effective defense against single-word substitutions and then extends this approach to handle multi-word substitutions.

One-word substitution: It describes the case that the attacker selects a word w_i in the text and replaces it with another word w_i from a set of synonyms. To defend against the attack with a probability t, it is necessary to select the smallest Gaussian noise standard deviation δ_{min} such that the distance covered by the Gaussian noise is greater than or equal to the maximum perturbation caused by the substitution. Specifically, if the word w is highly vulnerable to attack, it should be protected by adding Gaussian noise with a larger variance.

Multi-word substitution: When the attacker replaces multiple words simultaneously, the list L records the positions of all substituted words, where the position of each replaced word w_i is marked as 1. For each word w_i to be substituted, it record the possible maximum perturbation $||\delta_{max}||$. If the Gaussian noise standard deviation σ_i can be chosen to meet certain conditions for all i, the attack will be successfully defended. These conditions depend on the number of substitutions $d(x, x')$, the maximum perturbation $||\delta_{max}||$ for each word w_i, and certain probability thresholds p_A and p_B.

The results indicate that, for the CNN model, the Text-RS method exhibits significantly higher classification accuracy compared to other adversarial attack methods. For instance, in the IMDB dataset, Text-RS achieves a classification accuracy of 85.1%, while the accuracy of other methods ranges from 4.4% to 80.2%. Compared with the traditional discrete method, the robustness training effect of Text-RS is improved.

3.4. Chinese Variation Graph Integration

This paper examines adversarial sample techniques specifically in the context of the Chinese language [10]. It investigates methods to enhance pre-trained language models, aiming to bolster their robustness against adversarial attacks on Chinese text. The enhancement may involve optimizing the model's training process or incorporating specific techniques to better counteract malicious attacks, thereby increasing the model's security and stability in practical applications.

The Chinese Character Variation Graph, annotated $G = (c_0, r_0, c_1), (c_2, r_1, c_3), ..., (c_i, r_m, c_j)$ is a set of Chinese character variation approaches. c_i is Chinese characters, functioning as attacked character or attack character. $\mathrm{r_j}$ is the attack method between the two words, such as the transformation in Pinyin, Visual, Character to Pinyin.

Chinese Variation Graph Integration(CVGI): CVGI is a technique designed to enhance the robustness of PLMs against adversarial attacks specifically targeting Chinese text.

The first step of CVGI is recognizing attacked tokens by utilizing language model probability. For each token w_i in the context $C = (w_1, w_2, \ldots, w_n)$, use the output $f_{w_i}(C)$ from a language model (like BERT) to compute the probability of that token, where V is BERT vocabulary:

$$P(w_i|C) = \frac{exp\left(f_{w_i}(C)\right)}{\sum_{w_j \in V} exp\left(f_{w_i}(C)\right)} \tag{5}$$

Following the identification of attacked tokens, CVGI proceeds to reconstruct the input sentence by appending a postfix generated from the detected adversarial paths. These paths, such as pinyin, visual, or character variations, are annotated with specific tags to allow the model to differentiate between various forms of perturbations. To mitigate noise and reduce computational complexity, CVGI employs a two-dimensional attention mask. This mask restricts attention to interactions between the reconstructed adversarial tokens and the corresponding attacked tokens, rather than permitting full cross-attention with the entire original sentence.

This approach enables PLMs to more effectively discern the correct adversarial paths, facilitating the injection of accurate information from the original tokens into the attacked content. As a result, the model's ability to resist adversarial attacks is significantly enhanced. The CVGI framework is broadly applicable to a range of Chinese Natural Language Understanding (NLU) tasks, and its architecture is compatible with most transformer-based PLMs. Empirical results show that CVGI effectively enhances the adversarial robustness of PLMs.

4. Conclusion

This paper presents a thorough review of adversarial attack and defense techniques in the text domain. As neural networks become increasingly prevalent in text processing, the investigation of adversarial examples has emerged as a vital area of research. Adversarial examples, through small perturbations, can significantly impact model predictions, revealing vulnerabilities in the robustness and security of neural networks. These challenges not only affect the practical application of models but also present theoretical challenges in model development.

Currently, adversarial example generation techniques in the text domain, such as TextFooler, generate adversarial examples by replacing words with synonyms. The introduction of Search Space detects key feature, further proposing SSIP and SSET to estimate the different attack methods. For multi-word substitution defense, techniques like Text-RS improve model robustness through adaptive noise adjustment. Additionally, the CHANGE method enhances defense against adversarial examples in Chinese contexts by integrating variation graphs, which stabilize pre-trained language models.

Future research should focus on improving both adversarial example generation and defense techniques while exploring more generalized and efficient defense strategies. Furthermore, the issue of adversarial example transferability needs additional exploration to more effectively evaluate and strengthen the security and robustness of models. Through continuous improvements in techniques and methodologies, the future will see more secure and reliable text processing systems in practical applications.

References

[1] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2013). Intriguing properties of neural networks. arXiv e-prints, arXiv:1312.6199. https://doi.org/10.48550/arXiv.1312.6199

[2] Khriji, L., Messaoud, S., Bouaafia, S., Ammari, A. C., & Machhout, M. (2023). Enhanced CNN security based on adversarial FGSM attack learning: Medical image classification. *2023 20th International Multi-Conference on Systems, Signals & Devices (SSD)*, 360–365. https://doi.org/10.1109/SSD58187.2023.10411241

[3] Lohit, S., Liu, D., Mansour, H., & Boufounos, P. T. (2019). Unrolled projected gradient descent for multi-spectral image fusion. *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 7725–7729. https://doi.org/10.1109/ICASSP.2019.8683124

[4] Deng, K., Peng, A., Dong, W., & Zeng, H. (2021). Detecting C&W adversarial images based on noise addition-then-denoising. *2021 IEEE International Conference on Image Processing (ICIP)*, 3607–3611. https://doi.org/10.1109/ICIP42928.2021.9506804

[5] Yang, X., Gong, Y., Liu, W., Bailey, J., Tao, D., & Liu, W. (2023). Semantic-preserving adversarial text attacks. *IEEE Transactions on Sustainable Computing*, 8(4), 583–595. https://doi.org/10.1109/TSUSC.2023.3263510

[6] Zhang, H., Xie, Y., Zhu, Z., Sun, J., Li, C., & Gu, Z. (2021). Attack-words guided sentence generation for textual adversarial attack. *2021 IEEE Sixth International Conference on Data Science in Cyberspace (DSC)*, 280–287. https://doi.org/10.1109/DSC53577.2021.00045

[7] Jin, D., Jin, Z., Zhou, J. T., & Szolovits, P. (2019). Is BERT really robust? A strong baseline for natural language attack on text classification and entailment. *AAAI Conference on Artificial Intelligence*.

[8] Zhan, P., Yang, J., Wang, H., et al. (2024). Rethinking word-level adversarial attack: The trade-off between efficiency, effectiveness, and imperceptibility. *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, 14037–14052.

[9] Zhang, Z., Yao, W., Liang, S., et al. (2024). Random smooth-based certified defense against text adversarial attack. *Findings of the Association for Computational Linguistics: EACL 2024*, 1251–1265.

[10] Xiong, Z., Qing, L., Kang, Y., et al. (2024). Enhance robustness of language models against variation attack through graph integration. *arXiv preprint*, arXiv:2404.12014. https://doi.org/10.48550/arXiv.2404.12014