ECAI 2023
K. Gal et al. (Eds.)

doi:10.3233/FAIA230347

Towards Trustworthy NLP: An Adversarial Robustness Enhancement Based on Perplexity Difference

Zhaocheng Ge[a;*], Hanping Hu[a] and Tengfei Zhao[a]

[a]Huazhong University of Science and Technology

Abstract. The vulnerability of artificial intelligence has emerged as a bottleneck, with adversarial attacks posing a significant threat to natural language processing. Although multiple defense mechanisms have been proposed, they often suffer from strict constraints, weak generalization, and low scalability. To address these challenges, we propose leveraging perplexity to quantify the difference between clean and adversarial examples based on the observation of numerous cases. We then statistically prove the substantial difference between them using Bayesian hypothesis testing. Subsequently, we develop an adversarial defense framework named UMPS, which contains two branches: "Uncovering the Mask"(UM) and "Perplexity-guided Sampling"(PS). UM utilizes a masked language model and Jaro-Winkler distance constraint to recover out-of-vocabulary words, while PS employs perplexity to locate the optimal sample within a convex hull which is constructed with integrated gradients. Theoretically, the proposed framework fulfills three requirements: effectiveness, universality, and portability. The experimental results demonstrate that UMPS effectively enhances the robustness of language models including BERT, against advanced attacks and outperforms three strong baseline methods. Furthermore, we conduct an instance analysis to illustrate how UMPS functions and what it outputs, an ablation study to support the validity and necessity of the two branches, and an post-hoc test on the difference in perplexity to explains the defense performance of our framework.

1 Introduction

Artificial intelligence (AI) has significantly transformed the computing paradigm across multiple domains. However, AI's intrinsic uninterpretability has given rise to concerns over security, which hinders its development. As one of the major threats, adversarial attack refers to the misguiding of an AI model by adding subtle perturbations to the input data. Since Papernot et al. [16] first investigated adversarial attacks from the image to the text domain, the study of adversarial examples in Natural Language Processing (NLP) has gained substantial attention. Researchers have developed various attacks to expose the vulnerabilities of state-of-the-art models like ChatGPT [23] as well as of security-critical applications including classification, translation, comprehension, and conversation [6].

In response, various robustness enhancement techniques (i.e., defense) have been developed to safeguard NLP models. Adversarial training [5] is the most widely utilized method, mixing adversarial examples with benign ones for data augmentation or constructing a distinctive regularizer. This technique is formulated as a min-max game to minimize the adversarial loss. Moreover, spelling correction can effectively rectify grammatical errors or misspellings which can be regarded as char-level perturbations crafted by adversaries. To counter word-level attacks, detection-based strategy have been widely adopted [15, 22]. However, all these methods are based on empirical evidence. To provide theoretical certificates of robustness, there are two mainstream techniques: Interval Bound Propagation (IBP) [7, 9] and Randomized Smoothing(RS) [28].

Despite the varying advancements in both empirical defenses and certified methods, they leave some shared limitations that need to be addressed. For example, adversarial training is generally computation- and time-consuming. It also heavily depends on the quality of crafted examples, leading to limited resistance against unknown attacks. Similarly, some detection-based approaches require complex analysis of the input data, which would add an additional burden to the target model. Moreover, spelling corrections perform poorly when facing practical attacks that typically involve word-level substitution. Besides, IBP-based methods are restricted to model architecture, continuous space, and loose bound. RS-based defenses usually suffer costly computation and strong constraints. These observations motivate us to develop a universal defense technique that can effectively resist powerful attacks without heavy cost.

In this paper, we propose three principles to enhance adversarial robustness: *Effectiveness*—improve the accuracy of model under attack; *Universality*—ensure the effectiveness against various realistic attacks; *Portability*—cost-saving and model-agnostic. Subsequently, we draw a conclusion that *adversarial example is significantly and substantially worse than the original in terms of perplexity*, based on the observation of various pairs of original/adversarial texts and supported by statistical evidence obtained through Bayesian hypothesis testing. We then develop a novel framework, UMPS (**U**ncover the **M**ask and **P**erplexity-guided **S**ampling). It can protecting models against adversarial Attacks and requires no complex computation including modification or re-training, rendering it promising for real-world applications.

Our contributions are summarized as follows:

1. We propose perplexity to quantify the difference between original and adversarial texts, and draw an important conclusion.
2. We introduce two complementary strategies to mitigate the perplexity difference and adopt them to develop a defense framework named UMPS.
3. We evaluate UMPS through extensive experiments on two datasets, three models, and three attacks, which demonstrates that it outperforms three classic defense baselines.
4. We perform three analyses to validate the efficacy of UMPS, in-

* Corresponding Author. Email: gezhaocheng@hust.edu.cn

cluding an instance analysis of the output, an ablation study of its two branches and a post-hoc test on the perplexity difference.

2 Related Work

In response to the increasing threat of adversarial attacks, various techniques have been proposed to enhance the robustness NLP models. In this paper, we categorize these techniques into three groups: perturbation-oriented, model-oriented, and certified robustness.

Perturbation-oriented defenses aim to detect and recover from adversarial examples. One approach is to use a spelling correction module as a pre-processing step in front of a downstream NLP task to counter char-level attacks [17]. Besides, Wang et al. [25] propose Synonym Encoding Method (SEM) to defend against word substitution attacks by assigning a unique encoding to each cluster of synonyms. Zhou et al. [32] develop the DISP framework, which learns to discriminate perturbation to block both char- and word-level attacks. Mosca et al. [14] introduce a logits-based metric called Word-level Differential Reaction (WDR) to identify adversarial input texts. Mozes et al. [15] perform a statistical analysis of adversarial examples and developed a rule-based and model-agnostic algorithm, Frequency-Guided Word Substitutions (FGWS).

On the other hand, model-oriented defense is more proactive and focuses on the robustness of model itself. Adversarial training is the mainstream model-oriented method for achieving robustness and can be further categorized into two techniques: data augmentation and regularization. Data augmentation generates various adversarial examples and directly incorporates them into the training set, which is brute but effective. Regularization integrates perturbations into the training process and reformulates the cost function. For instance, Zhu et al. [33] add norm-bounded adversarial perturbations to the embeddings using a gradient-based method and minimize the resultant adversarial risk. Dong et al. [3] leverage Adversarial Sparse Convex Combination (ASCC) to capture perturbations inside the convex hull for adversarial training. Wang et al. [21] propose InfoBERT that contains two mutual-information-based regularizers for refining the local and global features. Moreover, some researchers enhance robustness by modifying NLP models. For example, SHIELD [11] patches the last layer of a well-trained neural network and transformed it into a stochastic weighted ensemble of multi-expert prediction heads. StaFF [29] is a novel fine-tuning framework that ensures both accuracy and stability of models under attack.

Different from the two empirical defenses, certified robustness aims to provide provable robust guarantees for a given model specification. In textual adversarial robustness, the common solutions can be categorized into Interval Bound Propagation (IBP) and Randomized Smoothing (RS). Specifically, Huang et al. [7] and Jia et al. [9] leverage IBP to compute an upper bound on the loss of the worst-case perturbation and minimize it to achieve certified robustness. Ye et al. [28] propose SAFER, a structure-free approach that constructs provable certification bounds based on the statistical properties of randomized ensembles. Zeng et al. [31] present a certified approach by randomly masking a proportion of the input words and discarding a common but unrealistic assumption. Wu et al. [26] leverage ranking and statistical property to achieve provable certification of top-K robustness. In addition to these two representative technical ideas, Xu et al. [27] demonstrate a dynamic programming approach to concretize linear bounds under discrete perturbations. Wang et al. [24] enhance robustness via differential privacy in text classification tasks and provide a rigorous analytic derivation of the certified condition.

3 Preliminary: Statistical Analysis of Adversarial Examples

3.1 Intuitive Observation on Generated Examples

Based on the limitations of existing defense methods, a new approach towards textual adversarial robustness is in need. We consider leveraging the differences between original and adversarial texts to investigate defense techniques. Therefore, we first need a large corpus of adversarial examples for observation. Instead of simple hand-crafted perturbations, five classic attacks are employed to generate adversarial texts.

Specifically, TextBugger [12] can generate adversarial texts that maintain their utility under both white-box and black-box settings. (2) DeepWordBug [4] introduces a novel scoring strategy and four subtle character manipulation. (3) Genetic [1] develops a black-box population-based optimization algorithm to minimize semantic and syntactic dissimilarity. (4) PWWS [18] is a popular synonym replacement attack based on Probability Weighted Word Saliency. (5) SememePSO [30] approaches attacking as a combinatorial optimization problem, incorporating the sememe-based word substitution and particle swarm optimization-based search. These five works are widely recognized as state-of-the-arts techniques at different periods of time, with over 2200 citations in total according to Google Scholar. They also cover a wide range of characteristics including char- and word -level perturbations. Therefore, they are eligible to be used as representative attacks for generating adversarial examples.

The five attacks are performed on two popular classification datasets, namely Stanford Sentiment Treebank (SST-2) and IMDB reviews, both of which are widely used in NLP research. 1000 benign texts are randomly selected from each dataset to generate adversarial examples. Additionally, three classic neural network, namely CNN, BiLSTM, and BERT, are considered as targets for the attacks. More details of datasets, models, and attacks can be found in Section 5.1.

We perform 30 sets of tests using two datasets, three models, and five attacks, which successfully generates 20,981 adversarial texts. Our findings reveal that although these generated texts can effectively deceive the model and retain the original semantics, a noticeable reduction in fluency and contextual relevance can be observed in most cases. However, these observations are inadequate for establishing a theoretical foundation, and more precise indicators are required to quantify and distinguish these characteristics among the examples.

3.2 Hypothesis Testing on Perplexity

Inspired by the evaluation of language model, we propose perplexity to assess the quality of texts, which can be understood as a measurement of uncertainty that is consistence with the characteristics observed above. Given a tokenized text $T = \{t_1, t_2, \ldots, t_k\}$, the perplexity of T can be calculated using Equation 1. It turns out that the perplexity value is equivalent to the exponentiation of cross-entropy. Our findings suggest that perplexity is an ideal measure of text quality, with lower values indicating more coherent and grammatically correct texts. In this paper, we prefer to calculate perplexity using GPT-2[1], a 1.5-billion-parameter Transformer model trained on 40 GB of texts from WebText.

[1] https://huggingface.co/gpt2

Table 1. The paired samples testing: μ, σ, $|d|$, BF_{10} refer to mean value, standard deviation, absolute of Cohen's d statistic, and Bayes factor respectively. Note that $|d|$ and BF_{10} are calculated on the natural logarithm of perplexity, i.e., $log(PPL)$.

Perplexity		SST						IMDB					
		Original		Adversarial		$\|d\|$	BF_{10}	Original		Adversarial		$\|d\|$	BF_{10}
		μ	σ	μ	σ			μ	σ	μ	σ		
CNN	TextBugger	312	588	1148	2565	1.51	10^{231}	86.4	42.1	149.1	102.6	1.25	10^{202}
	DeepWordBug	324	584	832	573	1.60	10^{199}	89.3	46.2	145.8	94.3	1.87	10^{118}
	Genetic	308	575	1071	3803	1.25	10^{179}	86.3	42.0	148.3	104.0	1.42	10^{234}
	PWWS	310	577	852	3603	1.18	10^{165}	86.2	41.9	127.8	88.0	1.21	10^{193}
	SememePSO	340	649	1134	9998	1.16	10^{118}	88.4	46.4	119.2	71.5	1.32	10^{52}
BiLSTM	TextBugger	316	563	1057	2450	1.44	10^{196}	86.5	42.3	149.8	105.9	1.13	10^{172}
	DeepWordBug	309	544	833	572	1.68	10^{173}	90.2	48.4	142.9	91.8	1.91	10^{110}
	Genetic	307	573	960	3500	1.20	10^{165}	86.7	42.2	140.4	84.7	1.40	10^{227}
	PWWS	302	567	734	1475	1.15	10^{150}	86.9	42.3	125.5	79.4	1.16	10^{69}
	SememePSO	340	643	946	5333	1.10	10^{111}	88.2	40.9	118.9	62.6	1.26	10^{172}
BERT	TextBugger	324	602	1087	1513	1.66	10^{202}	86.2	42.0	218.8	183.0	1.31	10^{229}
	DeepWordBug	318	569	851	599	1.62	10^{143}	88.6	40.2	138.3	63.9	1.87	10^{77}
	Genetic	316	616	927	2212	1.33	10^{182}	86.9	42.8	162.9	97.2	1.56	10^{243}
	PWWS	317	618	793	2230	1.25	10^{151}	93.3	50.8	185.0	128.4	1.32	10^{69}
	SememePSO	353	672	872	2197	1.27	10^{115}	93.0	53.4	128.0	83.3	1.14	10^{28}

$$
\begin{aligned}
PPL(T) &= \sqrt[k]{\prod_{i=1}^{k} \frac{1}{P(t_i|t_1 t_2 ... t_{i-1})}} \\
&= P(t_1 t_2 ... t_k)^{-\frac{1}{k}} \\
&= e^{-\frac{1}{k} \log_e P(t_1 t_2 ... t_k)} \\
&= e^{H(T)}
\end{aligned} \tag{1}
$$

Where $H()$ denotes cross-entropy between the actual data and model predictions. In the meantime, following [15], we conduct a Bayesian hypothesis testing on perplexity to compare the distributions of paired samples(i.e., original and adversarial texts), to provide statistical evidence for our findings. The null hypothesis **H0** and the alternative hypothesis **H1** are stated as follows:

Hypothesis 0 *the adversarial text is no worse than the benign one in terms of perplexity.* $PPL(T_{adv}) \leq PPL(T_{ori})$

Hypothesis 1 *the adversarial text is worse than the benign one in terms of perplexity.* $PPL(T_{adv}) > PPL(T_{ori})$

Two indictors are adopted in our testing. Based on the paired examples, we use the Bayes Factor BF_{10} to quantify the significance level of each hypothesis. Additionally, we provide $Cohen's\ d$ for the comparisons, which denotes the effect size of between-group differences. However, the exponential distribution of perplexity does not satisfy the prerequisite of quasi-normal approximation. Therefore, we transform all the data by taking the natural logarithm of the values to calculate $Cohen's\ d$ and BF_{10}, instead of directly analyzing the value of perplexity. This transformation have no effect on the result of testing.

As shown in Table 1, a clear difference in the distribution of perplexity between the original and adversarial examples exists in all 30 sets of attacks. Higher mean perplexity values indicate greater uncertainty, while higher standard deviations suggest a greater dispersion of this uncertainty. There is no doubt that the texts generated by adversarial attacks are characterized by both higher perplexity and greater dispersion, compared to the original texts. Furthermore, for **effect size**, that is the magnitude of the difference in perplexity between the two types of examples, all cases have $|d|$>1. As interpreted by Cohen [2], all comparisons have a huge effect, implying a substantial difference in perplexity. Similarly, for **significance level**, as suggested by Jeffreys [8], BF_{10}>100 indicates extreme evidence for the alternative hypothesis. All 30 tests surpass this critical value, strongly supporting **H1**.

Table 2. The paired samples testing for the ensembles: V and S refer to variable pairs and sample sizes respectively.

Dataset	V/S	$\|Cohen's\ d\|$	BF_{10}
SST	Ori/11270 Adv/11270	1.307	∞
IMDB	Ori/9711 Adv/9711	1.221	∞
SST+IMDB	Ori/20981 Adv/20981	1.121	∞

While the above analyses are focused on different attacks and models, we also investigate the performance of ensembles. The results on the two datasets and their combination are presented in Table 2, showing a significant difference in perplexity and a great likelihood of accepting hypothesis **H1**. It should be noted that the "∞" on BF_{10}, indicating the significant difference between the two hypotheses, is attributed to the tremendous sample size. Overall, the statistical analysis provides compelling evidence supporting the validity of the alternative hypothesis, as indicated by the Bayes factor. Furthermore, the results demonstrate a considerable effect size, represented by Cohen's d statistic. These findings, combined with the intuitive observations outlined in Section 3.1, lead us to conclude that *the perplexity of the adversarial text is significantly and substantially worse than that of the original text,* i.e., $PPL(T_{adv}) \gg PPL(T_{ori})$.

4 Methodology: UMPS

4.1 Overview

Based on the above conclusion, we propose two defense strategies: "Uncovering the Mask" and "Perplexity-guided Sampling", and develop a defense framework called UMPS which can defend against diverse adversarial attacks by reducing the perplexity of the input text. As shown in Algorithm 1, Step 1-9 and 10-18 refer to UM and PS, respectively. UM neutralizes char-level modifications by leveraging contextual information from a Masked Language Model (MLM) and constraining the Jaro-Winkler distance. PS mitigates the impact of word-level substitutions by performing perplexity-based sampling

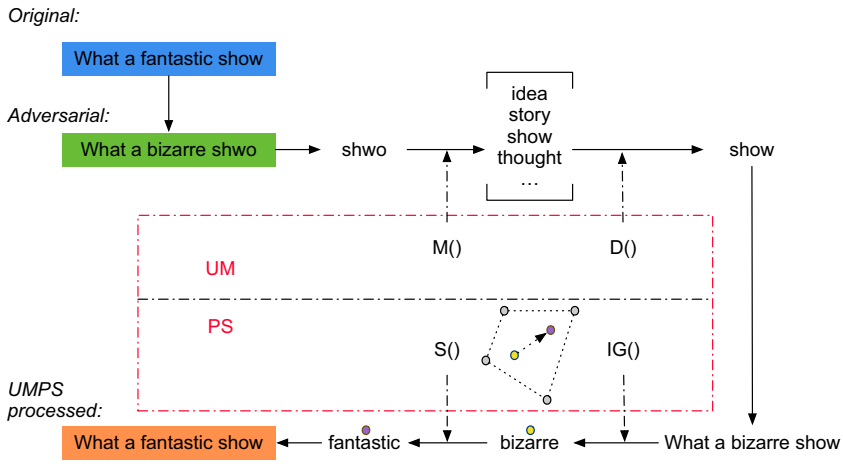

Figure 1. The workflow of UMPS: M(), D(), IG(), and S() refer to four key components respectively, which is elaborated in Section 4.

Algorithm 1 the workflow of UMPS

Input: $T = \{t_1, t_2, ..., t_k\} \leftarrow$ tokenized adversarial text
Params: $\mathcal{M}() \leftarrow$ masked language model
$WordNet() \leftarrow$ lexical database
$k, m, n \leftarrow$ length of T and two candidate sets
$l \leftarrow max(1, k//10)$
Output: $T_{umps} \leftarrow$ recovered text of T

procedure $UM(T_{masked})$
  <$mask$> $\leftarrow t_{adv_i} \leftarrow t_i$, if $t \in T$ is out-of-vocabulary
  $T_{masked} = \{t_1, ..., <mask>, ..., <mask>, ..., t_k\}$
  for <$mask$> in T_{masked} **do**
    $\{\dot{t}_1, ..., \dot{t}_m\} \leftarrow$ <$mask$>, by $\mathcal{M}(T_{masked})$
    $\hat{t} \leftarrow \{\dot{t}_1, ..., \dot{t}_m\}$, by Eq 2
  end for
  $T_{um} = \{t_1, ..., \hat{t}_{adv_i}, ..., \hat{t}_{adv_j}, ..., t_k\} \leftarrow T_{masked}$
end procedure
procedure $PS(T_{um})$
  $\{t_{adv_1}, ..., t_{adv_l}\} \leftarrow TopK(IG(t_i))$,$t_i \in T_{um}$, by Eq 3
  for each t_{adv} **do**
    $\{\dot{t}_1, ..., \dot{t}_n\} \leftarrow WordNet(t_{adv})$
    $\mathbb{C}(t_{adv}) \leftarrow \{\dot{t}_1, ..., \dot{t}_n\}$, construct a convex hull of t_{adv}
    $\hat{t}_{adv} \leftarrow \mathbb{C}(t_{adv})$, by Eq 4
    $T_{umps} \leftarrow \{t_1, ..., \hat{t}_{adv}, ..., t_k\}$, substitue $\hat{t}_{adv}$ for t_{adv}
  end for
end procedure
return T_{umps}

within a convex hull built in embedding space. Our framework offers comprehensive protection against adversarial attacks, does not require re-training and does not impose any burden on the target model. Additionally, it is model-agnostic that can be applied to any NLP model in any scenario. These features demonstrate that UMPS fulfills effectiveness, universality, and portability, which are the three principles proposed in Section 1.

4.2 Uncovering the Mask

Char-level attacks pose a significant threat to NLP applications by manipulating input texts, including swap, insert, delete, and substitute, which can reverse model predictions. These attacks often convert normal words into out-of-vocabulary words (OOVs), resulting in a significant decrease in perplexity. Despite the uniform replacement with <unk> by NLP models, OOVs can still have a considerable impact on predictions. Therefore, we propose a novel learning-based approach called "Uncovering the Mask" (UM) to recover OOVs into their original terms and reduce the perplexity of the input text.

Overall, UM first leverages powerful tokenizers such as Spacy and BertTokenizer to tokenize a text T into multiple tokens $\{t_1, t_2, \ldots, t_k\}$. We then utilize the GloVe model for distributed word representation and project these tokens into the word embedding space. We identify those tokens that do not exist in the vocabulary as OOVs and tag them with <$mask$> instead of <unk>. With the help of a Masked Language Model (MLM) and an edit distance constraint, UM can obtain the optimal recovery for each <$mask$> (i.e., OOV). Successful recovery of the original text indicates the failure of the adversarial attack.

In the process of "uncovering". BERT-based masked language models, which learn bidirectional representations, possess a natural advantage for the "fill-mask" task. RoBERTa, a member of the large BERT family, is preferred in our study to predict the masked words. In the default training, the model masks 15% of the tokens and then predicts them. We fine-tune the model on two datasets (SST-2 and IMDB) at a mask rate of 20%, which is an upper bound on the percentage of modifications for common attacks. Through 10 epochs of re-training, our masked language model can generate a set of candidate tokens for each <$mask$> in the input text.

However, MLM takes into account semantic features but not morphological information. To enhance the quality of the recovered tokens, we use the Jaro-Winkler Distance, an edit distance function, to search for the optimal token in the candidate set that is closest to the original one. This process is formalized as Equation 2. By combining MLM and JW, UM can effectively improve the perplexity and generate an semi-finished result.

$$\hat{t} = argmin\left(\left|D(t_{adv}, \dot{t}_i) - 1\right|\right), \quad t_{adv} \in T, i \in \{1, ..., m\} \quad (2)$$

where $D()$ is the Jaro-Winkler Distance function. The closer to 1, the higher the similarity; m is a hyper-parameter denoting the amount of candidate tokens, which is empirically set to 8 in our experiments.

4.3 Perplexity-guided Sampling

The aforementioned component is capable of mitigating character-manipulation-based attacks, but not word-substitution-based attacks that rarely produce grammatical errors including OOVs. Consequently, we develop a novel method named "Perplexity-guided Sampling"(PS). Given an text, we first use a saliency map to identify the key token and construct a convex hull of it in the embedding space. We then take an optimal sample, guided by the value of perplexity.

Specifically, constructing a convex hull for each token is computationally expensive, and successful attacks often rely on only a few substitutions. Inspired by [20], we leverage Integrated Gradients(IG) to build a saliency map and the calculation is formulated with Equation 3. Compared to other score- and attention-based saliency methods, IG is simple, accurate, and interpretable. We then extract the top-l tokens, where l is a hyper-parameter set at 10% of the text length (i.e., $l = max(1, k//10)$), since the average perturbation ratio of attacks is generally around 20%. These tokens are considered adversarial and have the most significant impact on the prediction.

$$IG_i(t) = \frac{t_i - t_i'}{m} \times \sum_{k=1}^{m} \frac{\partial F(t' + \frac{k}{m} \times (t - t'))}{\partial t} \tag{3}$$

where i represents the dimension; t' denotes the baseline which is usually an all-zero embedding vector of the same length as t; m is the number of steps that is defaulted as 100.

The attacks on language models often involve replacing a token with its synonym. In contrast, our method recovers the original token by identifying its synonyms. We first utilize WordNet to generate multiple synonyms for each adversarial token selected by IG, and accordingly construct a convex hull in embedding space. Given that adversarial examples are rare occurrences, it is reasonable to assume that the original token is included in this convex hull. However, a challenge arises in locating the original token.

Random sampling is a frequently used solution but it is less efficient and sometimes inaccurate. Gradient strategy has a drawback that defenders cannot determine whether the text is adversarial or clean. Therefore, we propose a perplexity-guided sampling method as an alternative, based on our previous finding that *the perplexity of the adversarial text is significantly and substantially worse than that of the original text*. Specifically, the best candidate is selected by ranking the perplexity of the sentences consisting of words in the convex hull, as stated by Equation 4.

$$\hat{t}_{adv} = argmin(PPL(T_{t_i})), \quad t_i \in \mathbb{C}(t_{adv}) \tag{4}$$

where $\mathbb{C}()$ denotes the convex hull in embedding space; T_{t_i} denotes the text made by substituting t_i for t_{adv}.

This candidate would be considered a reasonable substitution for the corresponding key token in the original sentence. We believe that the combined new sentence is equivalent to the original one. Note that this transformation is attack-agnostic. For an adversarial example, UMPS eliminates perturbations based on context and perplexity. While for a clean text, it can be regarded as a procedure of approximate expression. That is why UMPS can achieve successful defense without sacrificing the original model accuracy.

5 Experiment

5.1 Settings

Dataset: We make use of two typical datasets: Stanford Sentiment Treebank (SST-2) [19], and IMDB [13]. Both of them are provided by Stanford and used for text classification, with the main difference being the average length of the text. It should be noted that we randomly pick 1000 texts from the test set of each dataset to generate adversarial examples for statistical analysis and defense experiments. More details are shown in Table 3.

Table 3. The details of two datasets

	For model			For attack	
	Train	Dev	Test	Pos/Neg	Avg. length
SST	6920	872	1821	501/499	18.270
IMDB	40000	5000	5000	516/484	161.938

Model: Our experiments are tested on three representative architectures: (1) CNN is a convolutional neural network with a filter size of [3, 4, 5], 100 feature map, 1-max pooling, and ReLU activation function; (2) BiLSTM is a bi-directional recurrent structure with 2 LSTM layer and a hidden features dimension of 128. Both models are implemented based in PyTorch and trained from scratch with the same datasets and hyper-parameters (e.g., dropout rate=0.5, learning rate=1e-3, batch size=64); (3) BERT is a deep bidirectional transformer for language understanding. It is implemented based on "bert-base-uncased" provided by Hugging Face. We fine-tune it with the same settings as the first two models. Particularly, we adopt no additional optimization techniques, except for Adam.

Adversary: As with the previous process of generating adversarial examples, we consider multiple classic works as adversaries, instead of using simple hand-crafted perturbations. However, DeepWordBug and SememePSO achieved unsatisfactory success rates in the previous testing. Therefore, for our defense experiment, we only consider the other three, namely Genetic [1], PWWS [18], and TextBugger [12], which include both character- and word-level attacks. The introduction to the adversaries is given in Section 3.1 and we refer the interested reader there for detailed information.

Baseline: We compare our framework UMPS with three defense baselines that are no-retraining, model-agnostic, and widely referenced in the study of textual adversarial robustness: (1) ScRNN [17]: a robust word recognition model that integrates several backoff strategies on the ScRNN model and outperforms both adversarial training and off-the-shelf spell correction methods; (2) BERT-Defense [10]: a probabilistic model and an untrained iterative defense that combines context-independent and context-dependent information to find the best restoration in the space of sensible sentences; (3) FGWS [15]: a detect-then-recover strategy that exploits the frequency properties of adversarial substitutions and performs better than another similar and famous framework DISP [32].

Metric: We evaluate UMPS as well as the three baselines based on two metrics: (1) Model Accuracy represents the percentage of correctly classified texts, which is the most intuitive and crucial evidence in determining the defense capability of a safeguard; (2) Detection Success Rate refers to the percentage of successful defense texts out of the total number of successful attack texts.

5.2 Defense Performance

As presented in Table 4, our experiments show that all three attacks significantly reduce the model accuracy, whereas the four defenses improve the accuracy to varying degrees. For instance, on the experiment of {CNN, Genetic, SST}, ScRNN, BERT-Defense, FGWS, and UMPS increase the accuracy from 25.7% to 38.7%, 49.8%, 67.5%, and 71.9%, respectively. In general, UMPS surpasses the baselines in

Table 4. The accuracy of three language models: The optimal results have been bolded.

Accuracy/%	SST				IMDB			
	Original	Genetic	TextBugger	PWWS	Original	Genetic	TextBugger	PWWS
CNN	84.5	25.7	23.4	26.4	90.7	10.7	9.5	10.0
+ScRNN	81.8	38.7	71.9	39.4	**89.2**	45.9	80.8	41.8
+BERTDefense	80.7	49.8	73.6	52.6	87.6	61.7	83.0	53.0
+FGWS	80.8	67.5	37.4	72.4	86.9	71.2	49.8	77.7
+UMPS	**82.1**	**71.9**	**75.0**	**73.7**	87.4	**80.1**	**83.1**	**81.6**
BiLSTM	81.7	30.1	33.2	33.1	86.9	16.1	16.3	19.0
+ScRNN	**80.2**	41.4	73.1	42.4	**84.5**	51.5	77.6	49.6
+BERTDefense	78.2	49.7	72.6	52.8	83.7	52.4	79.1	61.3
+FGWS	79.4	64.8	50.5	69.6	82.1	79.8	57.4	**77.1**
+UMPS	79.8	**70.7**	**75.5**	**72.7**	84.2	**80.4**	**81.2**	71.5
BERT	90.1	25.0	28.1	33.5	93.6	14.9	14.8	54.3
+ScRNN	88.4	41.9	76.8	47.6	**93.2**	50.5	86.9	74.3
+BERTDefense	87.3	52.5	81.8	57.5	91.1	63.0	87.3	76.0
+FGWS	86.2	74.0	39.1	75.4	92.4	77.0	54.0	70.0
+UMPS	**89.6**	**80.3**	**83.1**	**82.3**	91.9	**84.1**	**88.3**	**79.6**

Table 5. The detection success rate of four defense methods: The optimal results have been bolded.

Detection/%	SST			IMDB		
	Genetic	TextBugger	PWWS	Genetic	TextBugger	PWWS
CNN+ScRNN	21.0	75.0	22.4	40.0	81.7	36.3
+BERTDefense	36.7	76.1	39.1	56.7	84.3	48.5
+FGWS	64.8	31.3	69.2	73.2	50.9	78.4
+UMPS	**70.7**	**78.5**	**73.7**	**81.9**	**87.8**	**85.4**
BiLSTM+ScRNN	18.0	72.0	18.1	44.0	79.4	38.3
+BERTDefense	33.7	72.4	34.8	49.3	80.1	67.7
+FGWS	69.4	51.2	69.9	70.9	61.6	**75.2**
+UMPS	**72.5**	**77.6**	**72.8**	**72.3**	**83.5**	71.9
BERT+ScRNN	25.5	75.4	24.3	42.6	91.7	34.3
+BERTDefense	40.6	82.3	39.4	58.8	**92.3**	45.3
+FGWS	76.0	41.9	81.4	77.4	56.9	60.8
+UMPS	**82.6**	**86.5**	**82.2**	**80.1**	84.6	**69.5**

terms of model accuracy, except for the result on {BiLSTM, PWWS, IMDB}. These findings support the effectiveness of UMPS in enhancing the robustness of NLP models.

Notably, ScRNN and BERT-Defense are proficient at defending against TextBugger, whereas FGWS performs better against Genetic and PWWS. We attribute this to variations in the universality of the defense methods. Genetic and PWWS primarily involve word substitution, while TextBugger incorporates five types of perturbations. The three strong baselines are only adapted to specific types of attacks. In contrast, UMPS performs more evenly when confronted with these advanced attacks, which provides evidence for the universality of our framework.

Furthermore, the performance on the original texts demonstrates that the four methods effectively preserve the accuracy of the model itself. In general, ScRNN and UMPS outperform the other two methods, albeit with negligible differences. It suggests that UMPS can successfully enhance robustness without compromising the accuracy, making it a suitable safeguard for various applications.

On the other hand, as shown in Table 5, UMPS also outperforms the three baselines in detection success rate. Despite being surpassed by other methods in 2 sets of experiments, our framework performs the best in the remaining 28 sets. It indicates that our framework can effectively detect and recover adversarial examples in various environment (with respect to data, attack, and model).

Overall, the results support the validity of the proposed defense framework for enhancing the adversarial robustness of NLP models.

6 Further Analysis

6.1 Instance Analysis

We present four real output instances on {BERT, PWWS, SST} in Figure 2. The results demonstrate that the language model accurately predicts labels for the original texts, whereas the subtly perturbed adversarial texts successfully mislead the model. Notably, when the adversarial texts are processed by UMPS, the predicted labels are correctly restored. This confirms the efficacy of UMPS in defending against adversarial attacks.

Then an important question arises: how does UMPS process the adversarial texts? To address it, we take T^2_{adv} as an example. During UM, we first identify the presence of an out-of-vocabulary token, "flim", and select "film" as the correction token from the candidate set {video, film, movie, ...}. Next, we determine the two keywords, "heed" and "creative", that have the most significant influence on the model. Subsequently, during PS, we generate the recovered text $T^2_{adv+umps}$. Specifically, in the corresponding candidate sets, {regard, mind, attention, ...} and {inventive, originative, innovative, ...}, "mind" and "innovative" are preferred for substitution.

Moreover, for the original texts, T_{ori} and $T_{ori+umps}$, UMPS functions more as an approximate expression procedure. Whether it is "sustain–support", "creative–innovative", or "process–operation", such approximations do not impact model predictions, which aligns with the defense performance observed in Table 4. We believe that the discrepancy caused by such approximations is inconsequential as

	Text	Pred Label	
		—	+UMPS
T^1_{ori}	there 's not enough to sustain the comedy	0	
$T^1_{ori+umps}$	there 's not enough to support the comedy		0
T^1_{adv}	there 's not enough to keep the comedy	1	
$T^1_{adv+umps}$	there 's not enough to support the comedy		0
T^2_{ori}	the best thing the film does is to show us not only what that mind looks like, but how the creative process itself operates	1	
$T^2_{ori+umps}$	the best thing the film does is to show us not only what that mind looks like, but how the innovative operation itself operates		1
T^2_{adv}	the best thing the flim does is to show us not only what that heed looks like, but how the creative process itself operates	0	
$T^2_{adv+umps}$	the best thing the film does is to show us not only what that mind looks like, but how the innovative process itself operates		1

Figure 2. The output instances of UMPS and their labels predicted by BERT: T_{ori}, T_{adv}, T_{umps} refer to the original, adversarial, and UMPS-processed texts, respectively; the red tokens are the ones recovered by UMPS; 1, 0 refer to positive and negative.

long as the model functions normally, since our objective is to enhance model robustness rather than recovering the text itself.

6.2 *Ablation Study*

In the previous experiments, UMPS is used as an ensemble. However, to explore the individual performance of the two branches, we conduct an ablation study where UM and PS are independently used to defend against attacks. Figure 3 shows that all three defenses can improve model accuracy under attacks to varying degrees. Specifically, UM is more effective in defending against TextBugger, while PS is better at Genetic and PWWS. Reviewing these attack algorithms, we find that TextBugger contains five perturbations covering character-level and word-level, while the latter two are dominated by word-level perturbations. This result demonstrates that UM and PS can defend against specific types of attacks individually. However, only by working together as UMPS can they achieve the best defense against those sophisticated attacks.

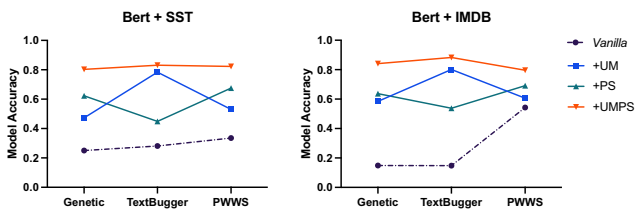

Figure 3. The accuracy of models under attacks when being protected by UM, PS and UMPS: *Vanilla* refers to the model without protection.

6.3 *Perplexity Test*

It is clear that UMPS is designed based on the difference in perplexity. To confirm the validity of it, we conduct a post-hoc analysis of the perplexity of texts in three different phases. As shown in Figure 4, the perplexity of the adversarial text is significantly higher than that of the original, which is consistent with the conclusion drawn in Section 3. In contrast, the perplexity of the UMPS-processed text is much lower than the adversarial text and is close to the original. The results suggest that our framework is indeed effective in reducing text perplexity as introduced in Section 4, which can explain the impressive defense capability of UMPS presented in Section 5.2.

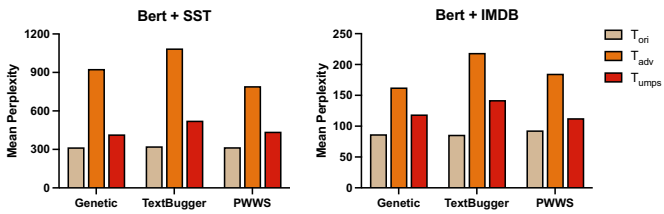

Figure 4. The mean perplexity of original, adversarial and UMPS-processed texts.

7 Conclusion

This paper presents UMPS, a novel defense framework designed to enhance the adversarial robustness of NLP models without additional re-training or modification on the architecture. It is characterized by its *effectiveness*, *universality*, and *portability*, making it suitable for various models. Our methodology is based on the finding that *the perplexity of the adversarial example is significantly and substantially worse than the original*, supported by intuitive observation and statistical analysis. Extensive experiments demonstrate the impressive performance of UMPS. For example, it improves the accuracy of TextBugger-attacked BERT from 14.8% to 88.3%, without compromising original accuracy. Overall, UMPS can serve as a promising patch for robustness enhancement. The discovery of the difference in perplexity between original and adversarial examples has important implications for the development of adversarial defense techniques.

We also identify some work to be continued, including exploring the application of UMPS in multimodal tasks and investigating the robustness of large language models. We intend to incorporate interpretable techniques for future research.

Acknowledgements

This work was partially supported by the Natural Science Foundation of China under Grant 61972448 and the Key R&D Program of Hubei Province under Grant 2020BAB104. Additionally, this paper is dedicated to Zhaocheng Ge, a Ph.D. candidate in CyberSecurity, in celebration of his forthcoming graduation. Wish him the best.

References

[1] Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang, 'Generating natural language adversarial examples', in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2890–2896, (2018).

[2] Sarah Cohen, Werner Nutt, and Yehoshua Sagic, 'Deciding equivalances among conjunctive aggregate queries', *Journal of the ACM*, **54**(2), (2007).

[3] Xinshuai Dong, Anh Tuan Luu, Rongrong Ji, and Hong Liu, 'Towards robustness against natural language word substitutions', in *International Conference on Learning Representations*, (2021).

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

[5] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy, 'Explaining and harnessing adversarial examples', in *3rd International Conference on Learning Representations*, (2015).

[6] Shreya Goyal, Sumanth Doddapaneni, Mitesh M. Khapra, and Balaraman Ravindran, 'A survey of adversarial defenses and robustness in nlp', *ACM Computing Surveys*, **55**(14s), (2023).

[7] Po-Sen Huang, Robert Stanforth, Johannes Welbl, Chris Dyer, Dani Yogatama, Sven Gowal, Krishnamurthy Dvijotham, and Pushmeet Kohli, 'Achieving verified robustness to symbol substitutions via interval bound propagation', in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pp. 4083–4093, (2019).

[8] Harold Jeffreys, *The theory of probability*, Oxford University Press, 1998.

[9] Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang, 'Certified robustness to adversarial word substitutions', in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pp. 4129–4142, (2019).

[10] Yannik Keller, Jan Mackensen, and Steffen Eger, 'BERT-defense: A probabilistic model based on BERT to combat cognitively inspired orthographic adversarial attacks', in *Findings of the Association for Computational Linguistics*, pp. 1616–1629, (2021).

[11] Thai Le, Noseong Park, and Dongwon Lee, 'SHIELD: Defending textual neural networks against multiple black-box adversarial attacks with stochastic multi-expert patcher', in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pp. 6661–6674, (2022).

[12] Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang, 'Textbugger: Generating adversarial text against real-world applications', in *26th Annual Network and Distributed System Security Symposium*, (2019).

[13] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts, 'Learning word vectors for sentiment analysis', in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150, (2011).

[14] Edoardo Mosca, Shreyash Agarwal, Javier Rando Ramírez, and Georg Groh, '"that is a suspicious reaction!": Interpreting logits variation to detect NLP adversarial attacks', in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pp. 7806–7816, (2022).

[15] Maximilian Mozes, Pontus Stenetorp, Bennett Kleinberg, and Lewis Griffin, '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*, pp. 171–186, (2021).

[16] Nicolas Papernot, Patrick McDaniel, Ananthram Swami, and Richard Harang, 'Crafting adversarial input sequences for recurrent neural networks', in *IEEE Military Communications Conference*, pp. 49–54, (2016).

[17] Danish Pruthi, Bhuwan Dhingra, and Zachary C. Lipton, 'Combating adversarial misspellings with robust word recognition', in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5582–5591, (2019).

[18] Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che, 'Generating natural language adversarial examples through probability weighted word saliency', in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1085–1097, (2019).

[19] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts, 'Recursive deep models for semantic compositionality over a sentiment treebank', in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, (2013).

[20] Mukund Sundararajan, Ankur Taly, and Qiqi Yan, 'Axiomatic attribution for deep networks', in *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pp. 3319–3328, (2017).

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

[22] Jiayi Wang, Rongzhou Bao, Zhuosheng Zhang, and Hai Zhao, 'Rethinking textual adversarial defense for pre-trained language models', *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, **30**, 2526–2540, (2022).

[23] Jindong Wang, Xixu Hu, Wenxin Hou, Hao Chen, Runkai Zheng, Yidong Wang, Linyi Yang, Haojun Huang, Wei Ye, Xiubo Geng, Binxin Jiao, Yue Zhang, and Xing Xie, 'On the robustness of chatgpt: An adversarial and out-of-distribution perspective', *arXiv:2302.12095*, (2023).

[24] Wenjie Wang, Pengfei Tang, Jian Lou, and Li Xiong, 'Certified robustness to word substitution attack with differential privacy', in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1102–1112, (2021).

[25] Xiaosen Wang, Jin Hao, Yichen Yang, and Kun He, 'Natural language adversarial defense through synonym encoding', in *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161, pp. 823–833, (2021).

[26] Chen Wu, Ruqing Zhang, Jiafeng Guo, Wei Chen, Yixing Fan, Maarten de Rijke, and Xueqi Cheng, 'Certified robustness to word substitution ranking attack for neural ranking models', in *Proceedings of the 31st ACM International Conference on Information and Knowledge Management*, pp. 2128–2137, (2022).

[27] Kaidi Xu, Zhouxing Shi, Huan Zhang, Yihan Wang, Kai-Wei Chang, Minlie Huang, Bhavya Kailkhura, Xue Lin, and Cho-Jui Hsieh, 'Automatic perturbation analysis for scalable certified robustness and beyond', in *Advances in Neural Information Processing Systems*, volume 33, pp. 1129–1141, (2020).

[28] Mao Ye, Chengyue Gong, and Qiang Liu, 'SAFER: A structure-free approach for certified robustness to adversarial word substitutions', in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 3465–3475, (2020).

[29] Zibo Yi, Jie Yu, Yusong Tan, and Qingbo Wu, 'Fine-tuning more stable neural text classifiers for defending word level adversarial attacks', *Applied Intelligence*, **52**(10), 11948–11965, (2022).

[30] Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun, 'Word-level textual adversarial attacking as combinatorial optimization', in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 6066–6080, (2020).

[31] Jiehang Zeng, Jianhan Xu, Xiaoqing Zheng, and Xuanjing Huang, 'Certified Robustness to Text Adversarial Attacks by Randomized [MASK]', *Computational Linguistics*, **49**(2), 395–427, (2023).

[32] Yichao Zhou, Jyun-Yu Jiang, Kai-Wei Chang, and Wei Wang, 'Learning to discriminate perturbations for blocking adversarial attacks in text classification', in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pp. 4904–4913, (2019).

[33] Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu, 'Freelb: Enhanced adversarial training for natural language understanding', in *International Conference on Learning Representations*, (2020).