TextVerifier: Robustness Verification for Textual Classifiers with Certifiable Guarantees

Siqi Sun and **Wenjie Ruan**⋆
University of Liverpool, Liverpool, UK
ssq@liverpool.ac.uk, w.ruan@trustai.uk

Abstract

When textual classifiers are deployed in safety-critical workflows, they must withstand the onslaught of AI-enabled model confusion caused by adversarial examples with minor alterations. In this paper, the main objective is to provide a formal verification framework, called *TextVerifier*, with certifiable guarantees on deep neural networks in natural language processing against word-level alteration attacks. We aim to provide an approximation of the maximal safe radius by deriving provable bounds both mathematically and automatically, where a minimum word-level L_0 distance is quantified as a guarantee for the classification invariance of victim models. Here, we illustrate three strengths of our strategy: *i)* ***certifiable guarantee:*** effective verification with convergence to ensure approximation of maximal safe radius with tight bounds ultimately; *ii)* ***high-efficiency***: it yields an efficient speed edge by a novel parallelization strategy that can process a set of candidate texts simultaneously on GPUs; and *iii)* ***reliable anytime estimation***: the verification can return intermediate bounds, and robustness estimates that are gradually, but strictly, improved as the computation proceeds. Furthermore, experiments are conducted on text classification on four datasets over three victim models to demonstrate the validity of tightening bounds. Our tool *TextVerifier* is available at `https://github.com/TrustAI/TextVerifer`.

1 Introduction

Deep neural networks (DNN) enable natural language processing (NLP) models to process human languages automatically and tackle various tasks, such as text classification (Garg and Ramakrishnan, 2020), machine translation (He et al., 2021), information retrieval (Tian et al., 2021), dialogue understanding (Zhong et al., 2022). Meanwhile, the robustness of textual classifiers requires to be ensured to prevent malicious attempts from adversaries due to inherent vulnerability(Mu et al., 2023; Wang et al., 2022; Zhang et al., 2023), which dates from computer vision (Szegedy et al., 2013; Goodfellow et al., 2015) and also emerges in NLP field (Jia and Liang, 2017).

Adversarial attacks (Ribeiro et al., 2018; Li et al., 2019; Jin et al., 2020) are designed to generate semantically or superficially similar outputs to fool the victim models. With regard to adversarial defence, multiple strategies are devoted to resisting adversarial attacks. *Adversarial training* (Cheng et al., 2020) is known as a mainstream mechanism to enhance the robustness of classifiers, which demands a substantial amount of adversarial examples for training. The shortcoming of this operation is the unavailability of all combinations with exponential growth among input sentences. Additionally, another effective scheme is *perturbation-based controlling* (Wang et al., 2021b), which aims to discern abnormal behaviors and rectify potential perturbations with safety risks. However, this scheme only specializes in dealing with visible perturbations, exclusive of unknown attacks. In that light, *robustness verification* is studied for solving the above problems. The main purpose is to provide a certificate for estimating robustness by identifying the local worst-case perturbations for a set of input sentences (Wu et al., 2020) .

In this paper, we mainly focus on performing local robustness verification by optimizing the problem of *maximum safe radius* (the minimum distance between an input and an adversarial example that changes the label prediction in Hamming space) against word-level substitution attacks, which possess strengths in semantic imperceptibility and syntactic fluency (Wang et al., 2019). Nevertheless, the computation process for such attacks has been shown to be an NP-hard problem (Zhai et al.). As shown in Figure 1, given a clean input sentence, our objective is to calculate the upper

⋆ Corresponding Author

Findings of the Association for Computational Linguistics: ACL 2023, pages 4362–4380
July 9-14, 2023

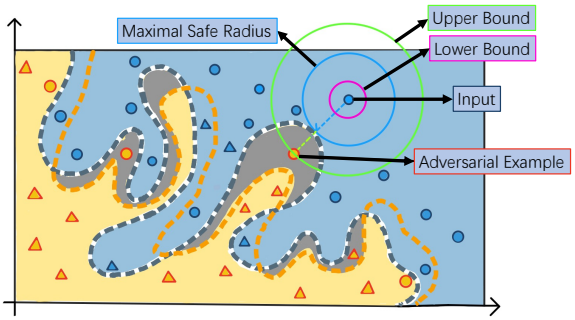


Figure 1: *An example for bounding the maximal safe radius in textual classification. The blue and yellow regions (blue dots and blue triangles: correctly and incorrectly classified samples, and vice versa for yellow samples) represent binary classification regions divided by a dark-blue dotted line, while the grey region means the human-perception boundary for the blue region separated by an orange dotted line. For the maximal safe radius, the upper bound (green line) is calculated by measuring the Hamming distance between the adversarial text (yellow triangle) and the input sentence (blue bold dot) if an adversarial example succeeds in misleading a textual classifier. The lower bound (pink line) represents that within this region, any perturbations are considered as safe while no misclassification appears. The maximal safe radius (blue solid line) is guaranteed to be approximated by the convergence between the upper and lower bounds eventually.*

and lower bounds to approximate the maximal safe radius with convergence guarantees and to process sets of potential adversarial sentences simultaneously via our efficient parallelization strategy. In our strategy, *the lower bound* ensures the nonexistence of adversarial examples within this safe region (Peck et al., 2017). While the *upper bound* is computed by any generated perturbations that result in incorrect predictions for NLP models, noted that the upper bound is utilized to approximate the safe radius from the upper side and localise the provably minimally-distorted adversarial example if it converges to the maximum safe radius (Carlini et al., 2017). Our approach verifies that the model is robust when the perturbations are below the lower bound (i.e., proving the robust region) and non-robust when the perturbations are above the upper bound (i.e., proving the non-robust region), leaving only the region between the lower and upper bounds unverifiable in terms of robustness. Iteratively, our method increases the lower bound and decreases the upper bound to narrow the uncertainty region, eventually *converging* to the maximum safe radius with *provable guarantees*. Our verification is an *anytime* algorithm. Namely, even if our tool *TextVerifier* is interrupted before the verification process is complete, it can still provide valid lower and upper bounds of the maximal safe radius for certifying the robust and non-robust region. The main contributions are listed below.

i) We propose an *effective* and *efficient* framework for verifying the robustness of textual classifiers against word-level alteration-based adversarial perturbations via maximal safe radius computation. Our mechanism provides *provable guarantees* regarding the correctness and bound convergence.

ii) Our novel *parallelization* strategy enables *high efficiency* in processing potential adversarial sentence sets simultaneously, which is applicable to efficiency-demanding scenarios.

iii) We employ an *anytime estimation* for a controllable and flexible approximation of maximal safe radius via bound computation within an acceptable time. As the computation proceeds, the upper and lower bounds will eventually converge to the maximal safe radius with provable guarantees.

iv) The upper bound calculated from our algorithm is demonstrated to be a competitive adversarial example crafting mechanism in terms of both attacking *efficiency* and *effectiveness*.

2 Related Work

Our work focuses primarily on robustness verification against word-level alteration-based attacks. So far, multiple representatives have been proposed based on synonym substitution, including (Alzantot et al., 2018; Ren et al., 2019; Jin et al., 2020; Li et al., 2020). Regarding realizing certified robustness by adversarial training, minimizing convex functions over convex sets is known as a convex optimization approach. In this line, WordDP (Wang et al., 2021a) was proposed to realize certified robustness against word-level substitution adversaries using differential privacy in text classification. Subsequently, a robust defence method called RanMASK (Zeng et al., 2021) was proposed based on randomized smoothing. It randomly masks a portion of words to avoid the hypothesis of the access to synonym candidate sets. In the work (Singla and Feizi, 2020), a Curvature-based Robustness Certificate (CRC) was proposed to compute a robustness certificate against L_2-bounded attacks by utilizing convex optimization. Nevertheless, it is inefficient to perform adversarial training via data augmentation for certified robustness.

In terms of robustness verification against adversarial attacks, Interval Bound Propagation (IBP)

is known as an incomplete method and was first proposed in computer vision (Gowal et al., 2018). Following this, (Jia et al., 2019) deployed IBP to the NLP settings, which considered the certified robustness of synonym substitution. Meanwhile, (Huang et al., 2019) introduced verifiable training to NLP models in the embedding space and computed tighter bounds. Then (La Malfa et al., 2020) measured robustness via maximal safe radius in the embedding space. However, the above strategies only focused on the embedding space has limited its application scenarios and IBP is difficult to extend to large network architectures such as BERT with a loose outer bound.

In summary, although robustness verification for NLP classifiers has been explored in previous research, existing approaches are either *computationally expensive* or cannot verify robustness in an *anytime* manner for *Hamming distance* with strict *convergence guarantees*.

3 Preliminaries

We start by providing *supporting notations* and *problem formulations*. The strategy for *robustness verification* will be detailed in the next section. Given an input sentence $X = \{x_i \mid i \in \mathbb{N}^*_{i\leq n}\}$ with n word-level tokens x. Then a pre-trained m-way NLP classifier f performs a mapping from the input X to a prediction label Y, i.e., $f : X \rightarrow Y$, where the label set is symbolized as $\mathcal{Y} = \{Y_j \mid j \in \mathbb{N}^*_{j\leq m}\}$ with m labels. Assume that C refers to the classification probability (i.e., a confidence score). From this, it follows that the prediction label Y for an input X is allocated via the highest confidence score, i.e., $f(X) = \arg\max\left\{C_j(X) \mid j \in \mathbb{N}^*_{j\leq m}\right\}$, and the ground-true label is denoted as Y^*.

Definition 1 (Problem of Maximal Safe Radius) For an input sequence X, a perturbation safe radius δ, a distance function $\|\cdot\|_k$ with $k \geq 0$, and a textual classifier $f : X \rightarrow Y$, a norm ball $\mathcal{B}(X, \delta, k)$ denotes a subspace that holds:

$$\mathcal{B}(X,\delta,k)=\left\{X_{\text{adv}} \middle| \|X-X_{\text{adv}}\|_k \leq \delta\right\} \quad (1)$$

where δ is defined as the maximum radius surrounding a given input X, ensuring the nonexistence of adversarial examples within the radius of a safe norm ball $\mathcal{B}(X, \delta, k)$.

Specifically, a norm ball $\mathcal{B}(X, \delta, k)$ is regarded as safe as formalized below:

$$\text{Safe}(\mathcal{B}(X,\delta,k)){:}\forall\ X_{\text{adv}}\in\mathcal{B}(X,\delta,k)\ \text{s.t.}\ f(X)=f(X_{adv}) \quad (2)$$

where a norm ball $\mathcal{B}(X, \delta, k)$ with the radius δ includes all possible sentences X_{adv} whose distance from X, measured by $\|\cdot\|_k$, is less than or equal to δ. Intuitively, if for any perturbed sentence X_{adv} within the norm ball $\mathcal{B}(X, \delta, k)$, the prediction invariance always holds, i.e., the prediction label remains unchanged: $f(X) = f(X_{\text{adv}})$, we say the norm ball is safe.

Hamming Space Our work sets $k = 0$ for the distance metric $\|\cdot\|_k$, which is known as the L_0-norm or Hamming distance. The Hamming distance between a clean input and an adversarial counterpart quantifies nonidentical words. Hamming distance (Ruan et al., 2019) can be an essential distance metric in the NLP field to measure the imperceptibility due to the discreteness of textual data. Meanwhile, the property of semantic fluency and syntactic naturality is preserved as our strategy is against word-level alteration-based attacks (Alzantot et al., 2018; Ren et al., 2019; Jin et al., 2020; Li et al., 2020). We further provide the definition of local robustness verification.

Definition 2 (Verification for Local Robustness) Given a clean text X, a model f with m labels, the confidence score C, the verification of local robustness $\text{Verify}(X)$ can be represented as:

$$\text{Verify}(X){:}\exists X_{adv}{:}\min \|X-X_{adv}\|_0\ \text{s.t.}f(X)\neq f(X_{adv}) \quad (3)$$

where we aim to obtain a minimum Hamming distance between the input X and the adversary X_{adv}, which is sufficient to lead to an incorrect prediction for the classifier f. This *minimum Hamming distance* can be regarded as *the maximum safe radius* if its minimality is certifiable.

The problem of robustness verification is actually an NP-hard problem due to the discreteness of the textual input and the non-convexity of the deep neural networks (Dong et al., 2020; Ruan et al., 2018). This paper aims to tackle this challenging problem by providing upper and lower bounds on this maximum safe radius, with a provable guarantee that the upper and lower bounds will optimize as the computation proceeds and eventually converge to the maximum safe radius. The key novelty of our solution lies in the *anytime* manner, i.e., whenever interrupting the computation, our method can return certifiable upper and lower bounds for radius approximation, as illustrated in Figure 1.

4 Robustness Verification

In this section, we present the key operations (*Stages 1-5*) and pivotal procedures (*lower and up-*

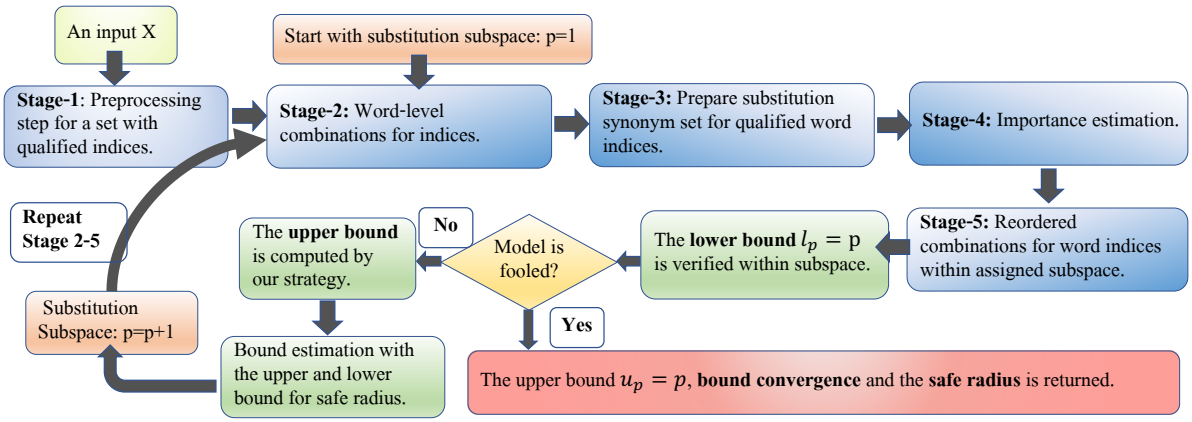


Figure 2: *The flowchart of the main stages for model verification with bound computation and safe radius estimation.*

per bound computation) for robustness verification, as shown in Figure 2. The details of our *anytime mechanism* for approximation-based estimation of the maximal safe radius will be explained in the next section.

4.1 Key Operations for Bound Computation

The flowchart depicting the key operations is shown in Figure 2. Stages 1-5 involve basic preparations for subsequent bound computation. Starting with an input sentence X, **Stage-1** preprocesses the input using common procedures for word-level alteration attacks. Then, by initializing the dimension of substitution combinations as $p = 1$, **Stages 2-3** generate combinations of substitution indices within the p-dimensional subspace and provide the corresponding synonym set for each alteration index. In **Stages 4-5**, the combinations of word indices are reordered based on importance estimation.

During the **bound computation**, the *lower bound* l_p is first verified based on the aforementioned p-dimensional substitution order to check whether the model is fooled. If the model is fooled, the upper bound u_p is set to p, indicating bound convergence and the safe radius is returned. Conversely, if the model is not fooled, the *upper bound* is computed using our strategy based on Stages 1-5. The *bound estimation* for a safe radius is obtained by considering the upper and lower bounds. The process continues iteratively by incrementing the substitution subspace dimension $p = p + 1$, and both the upper and lower bounds become stricter until the model is fooled, indicating *bound convergence*. Further details will be provided below.

Stage 1: Preprocessing Step for Qualified Indices In this stage, a common operation in word-level substitution attacks (Jin et al., 2020) is performed to filter out meaningless words using the NLTK library[2]. Recall that the input sequence $X = \{x_i \mid i \in \mathbb{N}^*_{i\leq n}\}$ consists of n tokens x, and the set of indices can be represented as $\{i \mid i \in \mathbb{N}^*_{i\leq n}\}$. Before mapping these tokens to embedding vectors, indices corresponding to insignificant words in the clean sentence are ignored and removed from the set of candidates to be substituted. The qualified word indices in the input X are sorted to form a subset $\mathbb{Q} \subseteq \{i \mid i \in \mathbb{N}^*_{i\leq n}\}$.

Stage 2: Word-level Combinations for Indices in Substitution Subspace Given a set $\mathbb{Q}$ includes qualified word indices to be substituted, assume that the potential adversarial example X_{adv} has modified p words simultaneously. Noted that p denotes the subspace dimensions for perturbation combinations. A 2-dimensional array is defined as $\mathbb{P}(\mathbb{Q}, p)$ under a p-dimensional alteration subspace, which consists of all possible combinations of word-level substitution indices originated from $\mathbb{Q}$ and the size of $\mathbb{P}(\mathbb{Q}, p)$ satisfies $\frac{|\mathbb{Q}|!}{(|\mathbb{Q}|-p)!} \times p$:

$$\mathbb{P}(\mathbb{Q},p)=\left\{\mathbb{P}_i \middle| \mathbb{P}_i \subseteq \mathbb{Q} \wedge i \in \mathbb{N}^*_{i\leq \frac{|\mathbb{Q}|!}{(|\mathbb{Q}|-p)!}}\right\} \quad (4)$$

where $\mathbb{P}_i$ means a subset of word indices in the universal set $\mathbb{P}(\mathbb{Q}, p)$ and $|\mathbb{P}_i| = p$. Therefore, given the indices set $\mathbb{P}(\mathbb{Q}, p)$, a set with all the potential adversarial sentences $X_{adv(\mathbb{P}_i)}$ is defined as $\mathrm{Sub}(X, p)$:

$$\mathrm{Sub}(X,p)=\left\{X_{\mathrm{adv}(\mathbb{P}_i)} \middle| \mathbb{P}_i \in \mathbb{P}(\mathbb{Q},p) \wedge i \in \mathbb{N}^*_{i\leq \frac{|\mathbb{Q}|!}{(|\mathbb{Q}|-p)!}}\right\} \quad (5)$$

where $X_{adv(\mathbb{P}_i)}$ means a single adversarial sentence with p word-level alterations based on the word indices subset $\mathbb{P}_i$ while the rest of word positions are unchanged.

[2] https://www.nltk.org/

Stage 3: Prepare Substitution Synonym Set for Qualified Word Indices In this stage, the synonym set for each original word index is allocated. Regarding synonym selection, the counter-fitting embeddings (Mrkšic et al., 2016) for vector space representations are utilized in our strategy, which has an edge in measuring semantic similarity, following the operations as (Jin et al., 2020). Recall that the input sentence $X = \{x_i \mid i \in \mathbb{N}^*_{i\leq n}\}$ with n words x_i. Therefore, a candidate set $\mathbb{W}_i$ with synonyms is created for an individual word x_i:

$$\mathbb{W}_i=\{\omega_{(i,j)}|i\in\mathbb{N}^*_{i\leq n},j=\mathbb{N}^*_{j\leq z}\} \quad (6)$$

where i denotes the word index in the input X and j denotes the synonym index in candidate set $\mathbb{W}_i$ for the original word x_i. Intuitively, the size of the set $\mathrm{Sub}(X,p)$ (Equation 5) is $\frac{|\mathbb{Q}|!}{(|\mathbb{Q}|-p)!}\times z^p$, consisting all the possible perturbed sentences.

Stage 4: Importance Estimation An array with importance scores for reordering substitution indices is obtained in this stage. Given an input X, a ground-true label Y^*, a confidence score C, an indicator function $\mathbf{1}_{\text{condition}}$, subspace dimension p, a universal set $\mathrm{Sub}(X,p)$ (Equation 5) with potential adversarial sentences X_{adv}, a set with importance scores corresponding to the set $\mathrm{Sub}(X,p)$ is defined as $\mathbb{S}(X,Y^*,p)$:

$$\mathbb{S}(X,Y^*,p)=\{[C_{Y^*}(X)-C_{Y^*}(X_{\text{adv}})]+ \mathbf{1}_{Y^*\neq Y_j}*[C_{Y_j}(X_{\text{adv}})-C_{Y_j}(X)]|X_{\text{adv}}\in\mathrm{Sub}(X,p)\} \quad (7)$$

where $\mathbf{1}_{Y^*\neq Y_j} = \begin{cases} 0 & \text{if } Y^*\neq Y_j \text{ is true,} \\ 1 & \text{if } Y^*\neq Y_j \text{ is false.} \end{cases}$ is an indicator function, which equals to 1 when the model misclassification appears. Otherwise, the function outputs 0 when the prediction label is true. Y_j denotes a different label from the ground-true label Y^*. Noted that the higher the importance score in $\mathbb{S}(X,Y^*,p)$ is, the greater possibility an adversarial sentence owns to fool the classifier. Specifically, $\mathbb{S}(X,Y^*,p) \in \mathbb{R}^{\frac{|\mathbb{Q}|!}{(|\mathbb{Q}|-p)!}\times z^p}_{\geq 0}$ is a $(p+1)$-dimensional array of importance scores.

Stage 5: Reordered Combinations for Word Indices within Assigned Subspace In this stage, local optimal combinations of synonym substitution are ordered based on $\mathbb{S}(X,Y^*,p)$ to resort the combinations of word indices. Given the array of importance scores $\mathbb{S}(X,Y^*,p) \in \mathbb{R}^{\frac{|\mathbb{Q}|!}{(|\mathbb{Q}|-p)!}\times z^p}_{\geq 0}$ (defined in Equation 4.1), a one-dimensional array with the highest importance scores along the first dimension with size z^p is represented as $\mathbb{S}^*(X,Y^*,p) = \{\max(\mathbb{S}_i(X,Y^*,p)) \mid i \in \mathbb{N}^*_{i\leq \frac{|\mathbb{Q}|!}{(|\mathbb{Q}|-p)!}}\} \in \mathbb{R}^{\frac{|\mathbb{Q}|!}{(|\mathbb{Q}|-p)!}}_{\geq 0}$. Here $\mathbb{S}^*(X,Y^*,p)$ represents the local optimal importance score for synonym substitutions along each input word. Therefore, the reordered set of combinations for original word indices is denoted as I_{word}:

$$I_{word}=\{\mathbb{P}_i(\mathbb{Q},p)|i\in\mathrm{argsort}_{\text{descend}}\,\mathbb{S}^*(X,Y^*,p)\} \quad (8)$$

where $\mathrm{argsort}_{\text{descend}}$ returns the original word indices with importance scores in descending order. $\mathbb{P}_i(\mathbb{Q},p)$ denotes a reordered subset of indices combinations.

4.2 Lower Bound Computation

The lower bound l_p is first verified within p-dimensional word-level subspace to check whether the model is fooled, if so, the upper bound u_p equals the lower bound, i.e., $u_p = l_p = p$. Otherwise, the upper bound is computed explained in the next subsection. As introduced in Definition 2, it is computational-consuming to calculate the maximum safe radius. Therefore, we aim to compute tight lower and upper bounds parallelly to alleviate query budgets and approximate the maximum safe radius in Definition 1 with provable convergence guarantees, as stated in the next section.

Given an input sequence X, a model f, recall that the size for $\mathrm{Sub}(X,p)$ (defined in Equation 5) is $\frac{|\mathbb{Q}|!}{(|\mathbb{Q}|-p)!}\times z^p$ stated in Stage 3. In a parallel setting, we query the textual classifier with $\mathrm{Sub}(X,p)$ directly for an array $\mathrm{Lab}(\mathrm{Sub}(X,p))$ with prediction labels, which is denoted as : $\mathrm{Lab}(\mathrm{Sub}(X,p)) = \{f(X_{adv}) \mid X_{\text{adv}} \in \mathrm{Sub}(X,p)\}$. Recall that X_{adv} is one of the potential adversarial sentences in the universal set $\mathrm{Sub}(X,p)$. The size of the array $\mathrm{Lab}(\mathrm{Sub}(X,p))$ is $\frac{|\mathbb{Q}|!}{(|\mathbb{Q}|-p)!}\times z^p$. This array is then compared with the ground-true label Y^* simultaneously, if any label is inconsistent with the original label, the lower bound for the maximal safe radius is regarded as $l_p = p$, while the upper bound equals to p and the safe radius converges. Specifically, the lower bound represents a safe radius without the existence of adversarial sentences. On the contrary, if all the labels in $\mathrm{Lab}(\mathrm{Sub}(X,p))$ remain coherent as the ground-true label Y^*, the lower bound is valued as p temporarily and increases iteratively.

4.3 Upper Bound Computation

As previously mentioned, if the model is not misled when verifying the lower bound l_p within p-

dimensional word-level subspace, the upper bound is computed as follows.

4.3.1 Processing Procedures

Similarly, our parallelization mechanism is also adopted in this part. Based on Stages 1-5, the reordered combinations of original word indices $I_{word}(X, p)$ in Equation 8 are utilized to match the local optimal synonyms. This step aims to obtain an order of local optimal synonym indices for substitution. Similarly, I_{syn} is created as a set including the synonym indices in the synonym set, which matches the local optimal synonym based on the array $\mathbb{S}(X, Y^*, p)$ with importance scores in descending order. The set is denoted as:

$$I_{syn}=\Big\{\arg\max\Big(\mathbb{S}_{(l,m)}(X,Y^*,p)\Big)|l=\{1,...,\tfrac{|\mathbb{Q}|!}{(|\mathbb{Q}|-p)!}\}, m=\{1,...,z^{q-1}\}\Big\} \quad (9)$$

where $\arg\max\big(\mathbb{S}_{(l,m)}(X, Y^*, p)\big)$ denotes an index in the synonym set. The indices are sorted by importance scores in descending order.

We start computing the upper bound by initializing the potential adversarial example X_{temp} as the input X. Then we iteratively add perturbations on X_{temp} based on I_{word} and I_{syn} within p-dimensional combinations of synonym-level modifications accumulatively. A potential adversarial example X^i_{temp} is generated by synonym substitution in ith iteration within p-dimensional subspace, originating from X^{i-1}_{temp} in last iteration:

$$X^i_{temp}=\{[X^{i-1}_{temp(I_{word(i,j)})}/\omega_{(I_{word(i,j)},I_{syn(i,j)})}]\} \quad (10)$$

where $[A/B]$ means replacing A (local optimal original word position to be substituted) with B (local optimal synonym in the synonym set). Specifically, $i \in \mathbb{N}^*_{i\leq|I_{syn}|}$ and $j \in \mathbb{N}^*_{j\leq p}$. Here, $I_{syn(i,j)}$ is the original word index for substitution in X^{i-1}_{temp} and $I_{syn(i,j)}$ is the synonym index in the synonym set for the original word. Moreover, the notation $\omega_{i,j}$ was defined in Equation 6, which represents an optimal synonym in Equation 10.

Furthermore, a set with potential adversarial examples X^i_{temp} is denoted as a set $\mathcal{A}(I_{syn})$ with $|I_{syn}|$ sentences, i.e., $\mathcal{A}(I_{syn}) = \{X^i_{temp} \mid i \in \mathbb{N}^*_{i\leq|I_{syn}|}\}$. Then we query the textual classifier with the set $\mathcal{A}(I_{syn})$ in parallel to acquire a set of classification labels $\mathrm{Lab}(\mathcal{A}(I_{syn}))$. In contrast to the ground-true label Y^*, if any sentence relates to X^i_{temp} results in an incorrect prediction, an adversarial example $X^i_{adv} = X^i_{temp}$ is generated with the optimal index i^*, which is returned from the upper bound of $i^* \cdot p$ word-level perturbations.

4.3.2 Redundancy Alleviation

To determine the maximal safe radius, we have calculated the Hamming distance between a clean input X and a perturbed sentence X_{adv} with the classification label changed as the upper bound. Although the obtained upper bound is tight, we can further tighten the bound by alleviating redundancy that exists in perturbed words that provide minor contributions to fooling NLP classifiers, i.e., $\|X_{adv} - X\|_0$ is optimized to a smaller scale.

Firstly, by creating a set I_{non} with indices by distinguishing nonidentical words between X_{adv} and X, a set R_{one} of adversarial sentences with a single-word substituted back to the original word is collected: $R_{one} = \{[X_{adv(I_{non(i)})}/x_{I_{non(i)}}] \mid x_{I_{non(i)}} \in X, i \in \mathbb{N}^*_{i\leq|I_{non}|}\}$. Then importance estimation is conducted by parallel querying the model with the set R_{one} for a set of reordered indices for substituting back in ascending order, denoted as I_{one} here, following the operations illustrated in Stage 4 and 5. In this line, a set R_{red} of accumulative word replacing-back sentences is generated by adopting I_{one} with size $|I_{red}| \times n$.

Simultaneously, the set R_{red} fed to the NLP classifier for a label set $\mathrm{Lab}(R_{red})$ and distinguish whether an identical label from ground-true label Y^* appears, the highest index j^* relates to the accumulative substitute-back sentence $X_{red(j)}$ is selected owing to the largest amount of perturbed words that are put back to the originals. Ultimately, an adversarial sentence has a total of j^* substitution-back words. This operation ensures a tight upper bound for convergence.

5 Anytime Estimation and Convergence Guarantees

We propose a safe-radius estimation method for obtaining an anytime robustness computation, ensuring the estimation results are controllable and flexible with convergence ultimately.

5.1 Anytime Estimation for Robustness

An anytime approximation is provided in a maximal safe radius estimation to ensure practical-scenario requirements. Instinctively, due to the access to the aforementioned bound interval $[l_t, u_t]$, the sample mean can be estimated by calculating: $\mathcal{M}(l_t, u_t) = \frac{l_t+u_t}{2}$. Moreover, the error bound can be computed by: $\mathcal{E}(l_t, u_t) = \frac{u_t-l_t}{2}$. In this line, an anytime robustness verification for safe ra-

dius, denoted as $\mathcal{R}(X,t)$ for a clean input text X within arbitrary acceptable runtime t, can be provided: $\mathcal{R}(X,t) = (\mathcal{M}(l_t,u_t), \mathcal{E}(l_t,u_t))$ and satisfies: $\mathcal{R}(X,t) \in [\mathcal{M}(l_t,u_t) - \mathcal{E}(l_t,u_t),\ \mathcal{M}(l_t,u_t) + \mathcal{E}(l_t,u_t)]$.

5.2 Bound Convergence and Certifiable Guarantees

Theorem guarantees and analysis of bound convergence are successively provided in this subsection, and corresponding proofs are given in Appendix.

Theorem 1 (Minimum Value Guarantee of Grid Search) Consider an input sentence X with q word positions to be substituted, a Lipschitz continuous NLP classifier f that is associated with a norm distance $\|\cdot\|_k$. The corresponding Lipschitz constant is set to L that holds $L \geq 0$. For each subspace dimension, assume that $\delta = \frac{1}{\Phi}$ words are selected recursively. Then a set with potential adversarial sentences is denoted as $\Gamma = \{X_{adv(1)}, \cdots, X_{adv(\delta^q)}\}$. We have: $\left\| f(X_{opt}) - \min_{X_{adv}\in\Gamma} f(X_{adv}) \right\|_k \leq L \cdot \left\| \frac{\Phi}{2} J_q \right\|_k$, where $f(X_{opt})$ denotes the optimal sentence results in a minimum value, $\min_{X_{adv}\in\Gamma} f(X_{adv})$ represents the minimum value obtained via grid search, and $J_q \in \mathbb{R}^{q\cdot q}$ symbolizes an all-ones matrix. The proof can be seen in *Appendix A.1*.

Theorem 2 (Lower Bound Computation Guarantee) Consider an input sequence X, an NLP classifier f for a m-way classification task, and the operation C for computing confidence scores. If for all the sentences X_{adv} that are perturbed within p-dimensional subspace word-level modifications, $f(X) = f(X_{adv})$ (i.e., the prediction label for X_{adv} remains unchanged) always holds, the lower bound $l_t = p$ ensures $\|X_{adv} - X\|_0 \leq l_t$ under runtime t. The proof is given in *Appendix A.2*.

Theorem 3 (Upper Bound Computation Guarantee) Consider a clean input X, an NLP classifier f and substitution subspace dimension $p \geq 0$ for bound computation, after being processed by our strategy, an upper bound u_p is created and optimized along with the growth of runtime. For all $p \geq 0$, the upper bound $u_p \geq u_{p+1}$ always holds. The proof is provided in *Appendix A.3*.

The above theorems provide theoretical guarantees for bound convergence. Intuitively, given the qualified indices number q, the uncertainty region between the upper and lower bounds is nonexistent when $p \mapsto q$ such that $\lim_{p\to q} \mathrm{Uncer}(l_t,u_t) = 0$, the proof is shown in *Appendix A.4*.

6 Experimental Part

As stated in *Section 2*, previous research on robustness verification was mainly focused on L_2 and L_∞ space, which is different from our L_0 norm setting. Moreover, since our method approximates the maximal safe radius from both lower and upper sides in a black-box manner, it effectively verifies the local robustness of large-scale, complex model structures such as large BERT models under L_0 norm. In this sense, it is difficult to find a verification baseline that has the same capability. Therefore, we evaluate the competitive tightness of upper bounds for approximating the maximal safe radius in *Subsection-6.2*. Moreover, we provide further experiments on both bounds in *Subsection-6.3* and *Appendix-B.4*, demonstrating the convergence performance for robustness verification.

6.1 Experimental Settings

1) Datasets: Four datasets are adopted for the task of textual classification, such as sentiment analysis and article multi-classification, to test the performance of our strategy: **i) MR:** A dataset for sentiment analysis with labels of positive and negative[3]. **ii) AG:** Multi-classification dataset for news reports. (Zhang et al., 2015)[4]. **iii) IMDB:** Binary sentiment analysis for movie reviews. (Maas et al., 2011)[5]. **iv) Yelp:** This dataset is created for binary polarity prediction (Zhang et al., 2015) [6]. More details can be found in *Appendix B.1*.

2) Model Settings: In text classification, three representative models are utilized for robustness verification: **i) BERT Base-uncased** (Devlin et al., 2019)[7], **ii) CNN** (Kim, 2014), **iii) LSTM**. The model descriptions can refer to *Appendix B.3*.

3) Representative Baselines: Previous research on robustness verification mainly focused on L_2, L_∞ norms, thus we adopt various representative attack strategies as baselines for verifying the tightness in terms of upper bound computation. Five word-level substitution attacks: **i) Genetic Attack** (Alzantot et al., 2018), **ii) Bert-Attack** (Li et al., 2020), **iii) PSO** (Zang et al., 2020), **iv) TextFooler** (Jin et al., 2020), **v) PWWS** (Ren et al., 2019), and a character-level attack **(vi) TextBug-**

[3] https://cs.cornell.edu/people/pabo/movie-review-data/

[4] http://groups.di.unipi.it/~gulli/AG_corpus_of_news_articles.html

[5] https://datasets.imdbws.com/

[6] https://www.yelp.com/dataset

[7] https://huggingface.co/textattack

Dataset	Attack	Bert			CNN			LSTM		
		Orig.%	Aveg. Val. ↓	Std. ↓	Orig.%	Aveg. Val. ↓	Std. ↓	Orig.%	Aveg. Val. ↓	Std. ↓
MR	**GA**	96.90%	2.95	1.83	86.60%	2.77	1.68	85.40%	2.80	1.84
	TextBugger		2.56	1.98		2.68	1.95		2.43	1.83
	PSO		3.39	3.38		2.63	2.57		2.40	2.44
	BERT-Attack		2.23	1.86		1.89	2.12		1.65	1.46
	PWWS		2.51	1.90		2.09	1.42		2.00	1.42
	TextFooler		3.15	2.56		2.14	1.27		1.99	1.36
	Ours		**1.57**	**1.00**		**1.46**	**0.70**		**1.46**	**0.72**
IMDB	**GA**	89.10%	9.02	7.16	89.20%	8.95	6.47	88.10%	8.39	6.44
	TextBugger		18.23	23.64		6.13	5.13		6.40	5.53
	PSO		4.87	6.26		5.02	4.28		4.75	4.34
	BERT-Attack		5.42	6.89		2.78	2.61		3.56	3.48
	PWWS		8.95	13.41		2.97	2.20		3.38	2.73
	TextFooler		15.55	19.05		3.70	2.52		4.03	3.16
	Ours		**4.50**	**3.79**		**2.12**	**1.00**		**3.10**	**2.43**
Yelp	**GA**	94.60%	9.39	6.69	94.00%	7.62	6.35	94.00%	6.80	5.16
	TextBugger		12.50	12.46		7.90	6.07		6.68	5.02
	PSO		6.67	4.58		5.58	4.73		5.35	4.10
	BERT-Attack		6.10	7.34		4.02	3.78		3.97	3.37
	PWWS		6.80	7.74		4.23	2.84		3.60	**2.42**
	TextFooler		9.59	10.09		4.68	3.01		4.00	2.64
	Ours		**4.12**	**2.58**		**3.55**	**2.79**		**3.58**	3.07
AG	**GA**	94.60%	5.96	3.22	89.90%	7.20	3.85	90.70%	5.98	3.90
	TextBugger		7.30	5.40		6.72	4.31		7.48	5.20
	PSO		9.22	5.28		7.60	3.98		7.76	4.24
	BERT-Attack		6.48	5.23		5.48	4.80		6.59	4.81
	PWWS		6.67	5.28		5.59	3.79		6.43	4.47
	TextFooler		8.82	6.17		5.87	3.61		6.93	4.70
	Ours		**3.67**	**3.06**		**2.72**	**1.58**		**3.76**	**2.71**

Table 1: When substitution subspace $p = 1$, the results of our method compared to six baselines (**2nd column**) over 4 datasets (**1st column**) for 3 classifiers (**1st row**). Abbreviations: original accuracy (**Orig.%**), the average value of upper bounds (**Aveg. Val.**), the standard deviation of upper bounds (**Std.**). ↓ means the lower, the better.

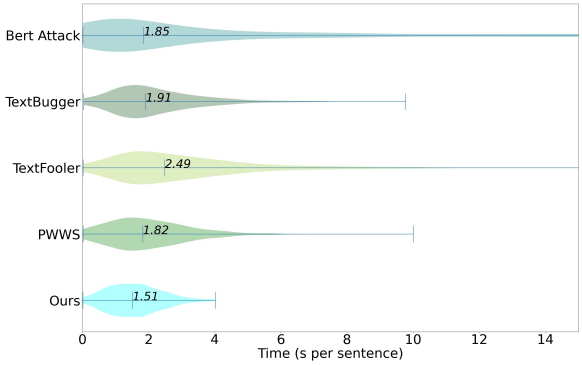


Figure 3: *Violin plot of computation runtime comparison for MR dataset with baselines regarding the computation of upper bounds.*

ger (Li et al., 2019) are selected and implemented on **TextAttack** framework (Morris et al., 2020). Baseline descriptions are detailed in *Appendix B.2*.

4) Evaluation Metrics: i) The average value and standard deviation of upper bounds: The average and standard deviation of Hamming distance between the original inputs and successful adversarial examples. **ii) Subspace dimension p:** The number of adopted substitution-based combinations for word-level alteration. **iii) Efficiency consideration:** The violin plot of runtime for the generation of each adversarial sentence.

6.2 Comparisons on Tightening Bounds and Computation Efficiency

By applying the L_0-norm to multiple NLP classifiers, we test the effectiveness of our proposed robustness-verification strategy in comparison with representative baselines in terms of calculating tight upper bounds, which can be seen in *Table 1*. According to the statistics in the table, our strategy compares with six baselines in terms of *average value* and *standard deviation of upper bounds*. Obviously, the lower the evaluation criteria are, the tighter the upper bound can be obtained. Therefore, our strategy has an edge in obtaining tight upper bounds to ensure faster convergence.

The computation runtime for each sentence under MR dataset performed on the BERT model is compared in the violin plot (*Figure 3*), our strategy has achieved a competitive efficiency as a verification mechanism, which ensures an anytime estimation of bounds. Note that PSO and Genetic Attack are not considered due to inherent weakness in low efficiency caused by the population-based algorithms. We perform the computation on a PC

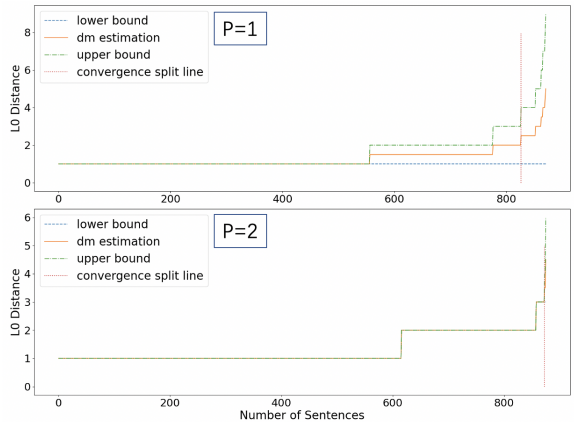


Figure 4: *When $p \in \{1, 2\}$, the upper bounds (green dotted line), lower bounds (blue dotted line), convergence split line (red dotted line) and estimations of maximal safe radius (orange solid line)* converges eventually.

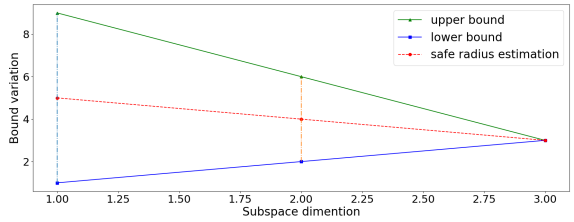


Figure 5: *Line chart of the bound convergence under CNN model for an individual sentence from the Yelp dataset when $p \in \{1, 2, 3\}$.*

with an i7-8559U CPU, 16GB RAM and NVIDIA RTX 2080Ti GPU.

6.3 Convergence Analysis for Upper and Lower Bounds

As shown in *Figure 4*, the convergence has been performed on the Bert model when subspace dimension $p = \{1, 2\}$. A total of 1000 clean sentences are selected, and almost 900 sentences can be classified correctly by the model. The sentences are sorted in ascending order based on the subspace dimension p. The estimation of the maximal safe radius and upper and lower bounds are shown in the figure. The red dotted line represents that all sentences to the left of this line have converged. When $p = 2$, we can see that the maximal safe radius of over 850 out of 900 sentences can converge. Therefore, in most cases, our algorithm converges after a few iterations for most sentences.

Within subspace dimension $p = \{1, 2\}$, we compute the lower bound, upper bound, and the maximum safe radius estimation for three models under four datasets (see *Appendix B.4*), which shows tighter bounds and convergence with the increasing iterative. The line chart for a random individual sentence under subspace dimension $p = \{1, 2, 3\}$ can be seen in *Figure 5*, when $p = 1$, the gap between the upper bound $u_p = 9$ and lower bound $l_p = 1$ is significant. With the increase of subspace dimension and runtime, the bounds eventually converge at 3 when $p = 3$.

7 Conclusion

The proposed strategy can verify the local robustness of NLP classifiers in Hamming space against word-level attacks with provable guarantees. A reliable anytime estimation effectively incorporates the computation of the upper and lower bounds to enable a controllable approximation of the maximal safe radius. Moreover, our framework is performed on a carefully-designed novel parallelization mechanism, ensuring its high efficiency. The extensive experiments demonstrate the competence of our method in obtaining very tight bounds for the maximal safe radius. Our *TextVerifier* tool also shows superior performance in terms of computational runtime, and provable convergence guarantees.

Limitations

In our work, when facing long sentences, a large number of synonym candidates can decrease the convergence speed of the lower and upper bounds. Therefore, in the experiments, we set up limitations on the length of the input sentences and the number of synonym candidates. Please note that it is still feasible to process long input sentences because of the *anytime* nature of our tool, however doing so would increase the unverifiable region, which essentially trades the tightness of bounds for efficiency.

Ethics Statement

Due to the vulnerability of textual classifiers to adversarial examples, here we highlight the potential ethical issues of word-level alteration-based attacks regarding misleading NLP classifiers in various tasks and escaping human observation via preserving semantic similarity and syntactic naturality. Our proposed strategy can tackle the above ethical issues by providing robustness verification against such attacks with provable guarantees and performing anytime approximation-based estimation for lower and upper bounds with high efficiency. Moreover, the algorithm for obtaining tight bounds has surpassed representative word-level substitution attacks in terms of perturbation rates. Therefore, our mechanism is competent in generating adversarial examples as well.

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.

Nicholas Carlini, Guy Katz, Clark Barrett, and David L Dill. 2017. Provably minimally-distorted adversarial examples. *arXiv preprint arXiv:1709.10207*.

Yong Cheng, Lu Jiang, Wolfgang Macherey, and Jacob Eisenstein. 2020. Advaug: Robust adversarial augmentation for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5961–5970.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186.

Xinshuai Dong, Anh Tuan Luu, Rongrong Ji, and Hong Liu. 2020. Towards robustness against natural language word substitutions. In *International Conference on Learning Representations*.

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.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. *stat*, 1050:20.

Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy Mann, and Pushmeet Kohli. 2018. On the effectiveness of interval bound propagation for training verifiably robust models. *arXiv preprint arXiv:1810.12715*.

Qiuxiang He, Guoping Huang, Qu Cui, Li Li, and Lemao Liu. 2021. Fast and accurate neural machine translation with translation memory. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3170–3180.

Po-Sen Huang, Robert Stanforth, Johannes Welbl, Chris Dyer, Dani Yogatama, Sven Gowal, Krishnamurthy Dvijotham, and Pushmeet Kohli. 2019. Achieving verified robustness to symbol substitutions via interval bound propagation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4083–4093.

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.

Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. 2019. Certified robustness to adversarial word substitutions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4129–4142.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8018–8025.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Emanuele La Malfa, Min Wu, Luca Laurenti, Benjie Wang, Anthony Hartshorn, and Marta Kwiatkowska. 2020. Assessing robustness of text classification through maximal safe radius computation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2949–2968, Online. Association for Computational Linguistics.

J Li, S Ji, T Du, B Li, and T Wang. 2019. Textbugger: Generating adversarial text against real-world applications. In *26th Annual Network and Distributed System Security Symposium*.

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.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150.

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.

Nikola Mrkšic, Diarmuid OSéaghdha, Blaise Thomson, Milica Gašic, Lina Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve

Young. 2016. Counter-fitting word vectors to linguistic constraints. In *Proceedings of NAACL-HLT*, pages 142–148.

Ronghui Mu, Wenjie Ruan, Leandro Soriano Marcolino, Gaojie Jin, and Qiang Ni. 2023. Certified policy smoothing for cooperative multi-agent reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI'23)*.

Jonathan Peck, Joris Roels, Bart Goossens, and Yvan Saeys. 2017. Lower bounds on the robustness to adversarial perturbations. *Advances in Neural Information Processing Systems*, 30.

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.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically equivalent adversarial rules for debugging nlp models. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Wenjie Ruan, Xiaowei Huang, and Marta Kwiatkowska. 2018. Reachability analysis of deep neural networks with provable guarantees. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 2651–2659.

Wenjie Ruan, Min Wu, Youcheng Sun, Xiaowei Huang, Daniel Kroening, and Marta Kwiatkowska. 2019. Global robustness evaluation of deep neural networks with provable guarantees for the hamming distance. International Joint Conferences on Artificial Intelligence Organization.

Sahil Singla and Soheil Feizi. 2020. Second-order provable defenses against adversarial attacks. In *International conference on machine learning*, pages 8981–8991. PMLR.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Yuanhe Tian, Guimin Chen, Yan Song, and Xiang Wan. 2021. Dependency-driven relation extraction with attentive graph convolutional networks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4458–4471.

Fu Wang, Chi Zhang, Peipei Xu, and Wenjie Ruan. 2022. Deep learning and its adversarial robustness: A brief introduction. In *HANDBOOK ON COMPUTER LEARNING AND INTELLIGENCE: Volume 2: Deep Learning, Intelligent Control and Evolutionary Computation*, pages 547–584.

Wenjie Wang, Pengfei Tang, Jian Lou, and Li Xiong. 2021a. 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*, pages 1102–1112.

Xiaosen Wang, Jin Hao, Yichen Yang, and Kun He. 2021b. Natural language adversarial defense through synonym encoding. In *Uncertainty in Artificial Intelligence*, pages 823–833. PMLR.

Xiaosen Wang, Hao Jin, and Kun He. 2019. Natural language adversarial attack and defense in word level.

Matthew Wicker, Xiaowei Huang, and Marta Kwiatkowska. 2018. Feature-guided black-box safety testing of deep neural networks. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 408–426. Springer.

Min Wu, Matthew Wicker, Wenjie Ruan, Xiaowei Huang, and Marta Kwiatkowska. 2020. A game-based approximate verification of deep neural networks with provable guarantees. *Theoretical Computer Science*, 807:298–329.

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.

Jiehang Zeng, Xiaoqing Zheng, Jianhan Xu, Linyang Li, Liping Yuan, and Xuanjing Huang. 2021. Certified robustness to text adversarial attacks by randomized [mask]. *arXiv e-prints*, pages arXiv–2105.

Runtian Zhai, Chen Dan, Di He, Huan Zhang, Boqing Gong, Pradeep Ravikumar, Cho-Jui Hsieh, and Liwei Wang. Macer: Attack-free and scalable robust training via maximizing certified radius. In *International Conference on Learning Representations*.

Chi Zhang, Wenjie Ruan, and Peipei Xu. 2023. Reachability analysis of neural network control systems. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI23)*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28:649–657.

Ming Zhong, Yang Liu, Yichong Xu, Chenguang Zhu, and Michael Zeng. 2022. Dialoglm: Pre-trained model for long dialogue understanding and summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11765–11773.

A Appendix

A.1 Proof of Theorem: (Minimum Value Guarantee of Grid Search)

Proof 1 Utilizing the Lipschitz condition for the NLP classifiers (Peck et al., 2017; Wicker et al., 2018), we demonstrate that grid search is a reliable strategy in approximating safe radius and locating the optimal solutions with provable guarantees when providing an error bound. Basically, an input sentence X is adopted in the theorems and proofs for clarity.

Given the knowledge of the Lipschitz continuity assumption for the NLP classifier f, the following condition is obtained:

$$\left\|f(X_1) - f(X_2)\right\|_k \leq L \cdot \|X_1 - X_2\|_k \quad (11)$$

The Φ-level grid search guarantees $\forall X,\ \exists X_{adv} \in \Gamma$ such that $\left\|X_{opt} - X\right\|_k \leq \left\|\frac{\epsilon}{2} J_q\right\|_k$. Thus the theorem holds as we can always find $\Gamma(X_{opt})$ from the set Γ for the minimum value of the sentence X_{opt}.

As demonstrated in Section 3.4, during each iteration, we utilize grid search to confirm the security of the NLP classifiers by ensuring that adversarial examples are prevented, based on a specified lower bound. By combining this approach with Theorem 2, we establish the following result, which provides a safety assurance for the lower bounds.

A.2 Proof of Theorem: Lower Bound Computation Guarantee

Proof 2 Due to the finiteness of the eligible words to be substituted in the sentence sequence for word-level alteration-based attacks, the Hamming distance between the adversarial sentence and the original input is limited as well. Therefore, as runtime t progresses, the lower bound l_t can be computed with a rising tendency and achieves convergence to the safe radius $\mathrm{rad}(X, f)$ ultimately. Therefore, our strategy for obtaining a lower bound l_t within runtime t for safe radius is guaranteed if, for any word-level perturbations within the lower bound l_t, model prediction invariance always exists.

This proof is demonstrated by *contradiction* based on Theorem 2 for the lower bound computation algorithm in Subsection 3.2. Consider an input sentence X, an NLP classifier f for a m-way classification task, confidence computation C, the ground-true label Y^*, the lower bound is denoted as l_t within computation runtime t and results in an adversarial sentence X_{adv} with the number of l_t word-level perturbations. Then assume that an adversarial example $X^{'}_{adv}$ with fewer perturbations than X_{adv} exits within $t^{'}$ computation runtime such that $t^{'} \leq t$ and the prediction label is changed: $Y_{X^{'}_{adv}} \neq Y^*$. Moreover, the number of modified words for $X^{'}_{adv}$ is represented as $l_{t'} = \left\|X - X^{'}_{adv}\right\|_0$ and satisfies $l_{t'} \leq l_t$.

Recall that the set with potential adversarial examples within p-dimensional subspace for the input X is denoted as $\mathrm{Sub}(X, p)$ defined in Equation 5 and $p \in \mathbb{N}^*_{p \leq |\mathbb{Q}|}$. Then we have $X_{adv} \in \mathrm{Sub}(X, l_t)$ and $X^{'}_{adv} \in \mathrm{Sub}(X, l_{t'})$. Due to the monotonic increasing order for value p, the label invariation for $\mathrm{Sub}(X, l_{t'})$ should be tested before $\mathrm{Sub}(X, l_t)$. Therefore, when $X^{'}_{adv} \in \mathrm{Sub}(X, l^{'}_t)$ and given that $Y_{X_{adv}} = Y^*$, there always exists $Y_{X^{'}_{adv}} = Y^*$. when the runtime $l_{t'} \leq l_t$. Here shows the contradiction with our original assumption that $Y_{X^{'}_{adv}} \neq Y^*$. Therefore, it is proved that the lower bound for safe radius approximation is computed with provable guarantees.

A.3 Proof of Theorem: Upper Bound Computation Guarantee

Proof 3 The computation for upper bounds satisfies the following requirements to provide the guarantees: firstly, given the subspace dimension p for substitution positions, as the runtime increases, the set of potential adversarial sentences created by adopting p-dimensional combinations is always the subset of the adversarial sequences under $(p+1)$-dimensional modification strategy. Secondly, an optimal candidate synonym for a specific original word always has an edge in causing the highest importance score variation and those optimal synonyms indicate the decreasing influence sequence. Thirdly, our redundancy strategy leads to computing a relatively tight upper bound eventually for faster convergence speed.

In this proof, *complete induction* is utilized by providing a base case and an inductive step, which aims to demonstrate the property of monotonic decreasing for tightening the upper bounds with the increase of the computation runtime.

Base Case $p = 1$: As stated in Subsection 3.2 for upper bound computation, assume that the scenario with subspace dimension $p = 1$ for substitution perturbation, an adversarial sentence X_{adv} is generated such that the prediction label changes: $Y_{X_{adv}} \neq Y^*$.

Recall that within p-dimensional substitution subspace, I_{syn} (defined in Equation 9) is a set of pairs for indices of the original word locations and local optimal synonyms based on the importance score array $\mathbb{S}(X, Y^*, p)$ and the set $\mathrm{Sub}(X, p)$ includes potential adversarial sentences. When substitution perturbation dimension $p = 1$, assume that an adversarial example has m words perturbed, i.e., a set with $\{I_{syn(1)}, I_{syn(2)}, \cdots, I_{syn(m)}\}$ constitutes the necessary modification indices for a misclassification such that the upper bound $u_p = m$. Then when the subspace dimension is $p+1$, the $p+1$-dimensional adversarial sequence set $\mathrm{Sub}(X, p+1)$ satisfies $\mathrm{Sub}(X, p) \subset \mathrm{Sub}(X, p+1)$ because an additional substitution dimension is considered, thus the highest confidence score that results in fooling the model satisfies: $\max\ \mathbb{S}(X, Y^*, p) \leq \max\ \mathbb{S}(X, Y^*, p+1)$. In terms of the Hamming distance under this case, we have $\left\|X_{adv(p)} - X\right\|_0 \geq \left\|X_{adv(p+1)} - X\right\|_0$, i.e., $u_p \geq u_{p+1}$ always holds.

Inductive Case: $p = k$: When performing $p = k$ dimensional perturbations, the set with adversarial sentences is denoted as $\mathrm{Sub}(X, k) = \{\mathrm{Sub}(X, 1) \cup \mathrm{Sub}(X, 2) \cup \cdots \mathrm{Sub}(X, k)\}$. While $\mathrm{Sub}(X, k) = \{\mathrm{Sub}(X, 1) \cup \mathrm{Sub}(X, 2) \cup \cdots \mathrm{Sub}(X, k)\} \cup \mathrm{Sub}(X, k+1)\}$. Similarly, via the progress for upper bound tightness, the set with potential sentences satisfies $\mathrm{Sub}(X, k) \subset \mathrm{Sub}(X, k+1)$ such that the upper bound holds $u_k \geq u_{k+1}$. Then the confidence score satisfies $\max\ \mathbb{S}(X, Y^*, k) \leq \max\ \mathbb{S}(X, Y^*, k+1)$ because the set $\mathbb{S}(X, Y^*, k+1)$ has a larger subspace to select optimal substitution combinations than $\mathbb{S}(X, Y^*, k)$. Therefore, with regard to the Hamming distance under k and $k+1$ dimensional substitution subspace, we have $\left\|X_{adv(k)} - X\right\|_0 \geq \left\|X_{adv(k+1)} - X\right\|_0$, i.e., the upper bound $u_k \geq u_{k+1}$.

A.4 Proof of Theorem: The Property of Uncertainty Region for Safe Radius Convergence

Proof 4 (Uncertainty Region with Bound Convergence Guarantees: $\lim_{p \to q} \mathrm{Uncer}(l_t, u_t) = 0$) Recall that the input is X, the NLP classifier is f, the number of eligible original words is q, the subspace substitution dimension is p, the radius estimation is $\mathcal{R}(l_t, u_t)$, the Hamming distance relation is denoted as $0 \leq l_t \leq \mathcal{R}(l_t, u_t) \leq u_t \leq q$, where l_t and u_t are the lower and upper bound within runtime t respectively.

Given a subspace dimension p, two arbitrary subspace dimensions a and b satisfy $a, b \in \mathbb{N}^*_{a,b \leq p \wedge a < b}$, assume that the lower bound and upper bound within p dimensions are denoted as l_p and u_p, if $l_a \leq l_b \wedge u_b \leq u_a$ always holds, the bound convergence is guaranteed at a certain time. Specifically, both the upper and lower bounds are approaching a convergence point more and more explicitly, the convergence point will be obtained eventually.

Following this, if the subspace dimension $p \to q$, we have the upper bound $u_t \to q$ and the lower bound $l_t \to q$. As defined in Subsection 5.1, the estimated safe radius in any time t: $\mathcal{R}(X, t) \in [\mathcal{M}(l_t, u_t) - \mathcal{E}(l_t, u_t),\ \mathcal{M}(l_t, u_t) + \mathcal{E}(l_t, u_t)] = [\frac{l_t + u_t}{2} - \frac{u_t - l_t}{2}, \frac{l_t + u_t}{2} + \frac{u_t - l_t}{2}]$, where $\mathcal{E}(l_t, u_t) = \frac{u_t - l_t}{2} = 1/2(q - q) = 0$ when $p \to q$. Therefore, the convergence property of the safe radius is guaranteed within finite word substitution positions, i.e.,$\mathcal{R}(X, t) = \mathcal{M}(l_t, u_t) = \frac{l_t + u_t}{2} = q$. Under this case, the uncertainty region (i.e., the region between the upper and lower bound) satisfies: $\mathrm{Uncer}(l_t, u_t) = u_t - l_t = q - q = 0$, which means the uncertainty region does not exist here.

B Appendix

B.1 Dataset Details

The descriptions and links for datasets are listed below. The label number, maximal sequence length in experiments, synonym number selection and the dataset number for model training can refer to Table 2.

(i) MR: A dataset originated from movie reviews for processing sentiment analysis with labels of positive and negative [8]. **(ii) AG:** A subset of AG's corpus that belongs to the category of news utilized for multi-classification. The prediction labels for those news reports cover world, sports, business and sci/tech (Zhang et al., 2015)[9]. **(iii) IMDB:** In comparison with previous benchmark datasets, this dataset contains considerably more textual data for the use of binary sentiment analysis (positive \ negative) (Maas et al., 2011)[10]. **(iv) Yelp:** This dataset is created for binary polarity (sentiment)

[8] ttps://www.cs.cornell.edu/people/pabo/movie-review-data/

[9] http://groups.di.unipi.it/~gulli/AG_corpus_of_news_articles.html

[10] https://datasets.imdbws.com/

prediction (Zhang et al., 2015) [11].

Dataset	MR	AG	IMDB	Yelp
Label Number	2	4	2	2
Max Length	100	50	50	50
Synonym Number	50	20	20	20
Train Set	9,000	120,000	25,000	560,000
Test Set	1,000	7,600	25,000	38,000

Table 2: The settings of datasets for sequence classification in our experiment, including label number, maximum sequence length selected in experiments, synonym number selection and the sentence number in training and testing stage.

B.2 Baseline Descriptions

The descriptions for baselines are stated below:

(i) Genetic Attack: A word-level population-based optimization method in a black-box setting for crafting adversarial sentences (Alzantot et al., 2018). **(ii) TextBugger:** A bug generation method by proposing character-level modification such as inserting, deletion, swapping and replacement (Li et al., 2019). **(iii) PSO:** A word-level adversarial attack that combines the concept of sememe (the minimum semantic unit of human languages) and particle swarm optimization-based search algorithm (Zang et al., 2020). **(iv) TextFooler:** This paper has considered performing word transformer and semantic similarity checking for generating adversarial attacks(Jin et al., 2020). **(v) PWWS:** A greedy algorithm is proposed by utilizing probability weighted word saliency for word substitution (Ren et al., 2019). **(vi) Bert-Attack:** The adversarial samples are constructed by deploying the BERT masked language model (Li et al., 2020).

B.3 Model Details

The details of adopted models are provided:

(i) BERT Base-uncased: A transformer-based model with 12 layers of the block, 768 hidden size, 12 self-attention heads and 110M trainable parameters [12]. **(ii) CNN:** The size of filters is 150, the dropout rate is set to 0.3 and the Adam optimizer is adopted. **(iii) LSTM:** A single-layer bi-direction LSTM is utilized with 150 hidden units. Similar to CNN, the dropout rate is 0.3 and the Adam optimizer is utilized.

B.4 The Bound and Estimation Comparison under Different Subspace Dimensions

As seen in Figure 6, within the substitution subspace $p = \{1, 2\}$, the average lower bound, the average upper bound and the safe radius estimation comparisons are depicted for three models (Bert, CNN and LSTM) under four datasets (MR, IMDB, Yelp and AG). This experiment is performed on 1000 sentences on each dataset and gets the average of bounds. When the value of p increases, both the lower and upper bounds are becoming more tightening. Especially for some of the models like Bert on MR dataset, we can see that when $p = 2$, the bound convergence appears.

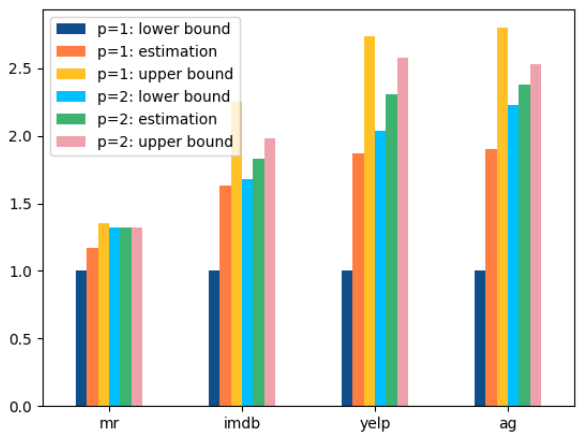
B.5 The Heatmap for Word Importance Scores

The example of a heatmap with word importance scores is shown in Figure 7, the sentence is selected from the MR dataset. After filtering out meaningless words, the qualified words can be seen in the first row. After word-level importance estimation, the synonym "establishing" with the darkest colour has gained the highest importance impact for substitution for the original word "creation". Then based on our strategy, the synonyms such as "singular" and "mesmerised" are the following substitution candidates based on a descending order for importance scores. For clarity, the importance score is normalized to [0,1].

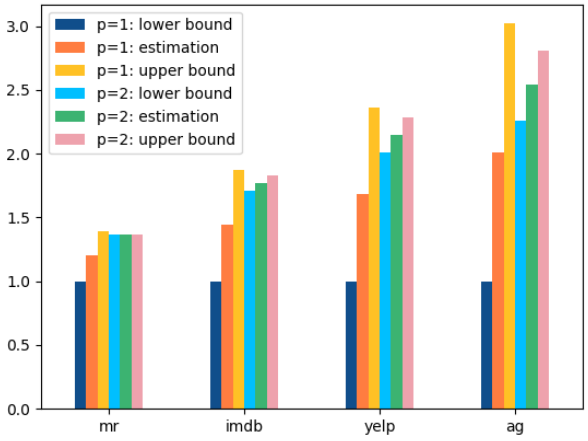
B.6 Samples of Adversarial Examples

In this subsection, successfully generated samples of adversarial examples are provided on both tasks of text classification and textual entailment in Table 3. This table involves the datasets of MR (binary sentiment analysis), AG (article multi-classification) and SNLI (textual entailment) for BERT and CNN models. As shown in the table, only a small quantity of words are perturbed with synonyms that can change the prediction labels for NLP models. This table demonstrates that our strategy can not only provide tight upper bounds but also be competitive in designing an algorithm for adversarial attacks.

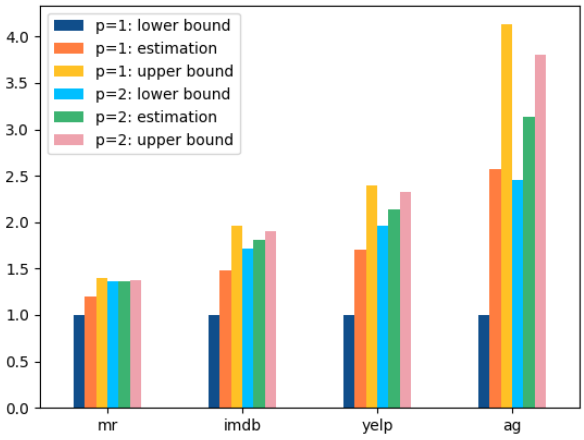
[11] https://www.yelp.com/dataset
[12] https://huggingface.co/textattack



(a) Bert



(b) CNN



(c) LSTM

Figure 6: When $p = \{1, 2\}$, the lower bound, upper bound and the safe radius estimation comparison for three models (Bert, CNN and LSTM) under four datasets (MR, IMDB, Yelp and AG).

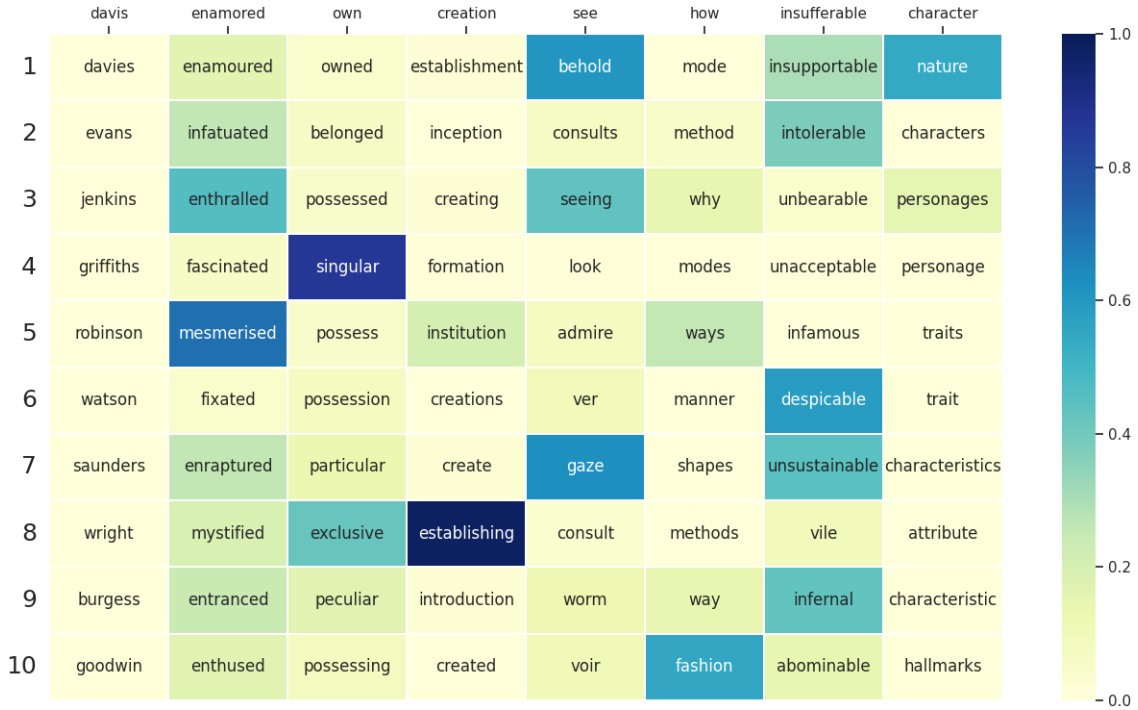


Figure 7: *The heatmap of a sentence in the MR dataset for word-level importance estimation. For clarity, the importance score is normalized to [0,1].*

	Rotten Tomatoes Movie Reviews (MR) for Bert (Labels: **Positive, Negative**)
Original (neg.)	An **annoying** orgy of excess and exploitation that has no point and goes nowhere.
Adversary (pos.)	An **unnerving** orgy of excess and exploitation that has no point and goes nowhere.
Original (pos.)	An hour and a half of **joyful** solo performance.
Adversary (neg.)	An hour and a half of **happier** solo performance
Original (pos.)	A muckraking job, the cinematic equivalent of a legal indictment, and a fairly **effective** one at that.
Adversary (neg.)	A muckraking job, the cinematic equivalent of a legal indictment, and a fairly **salubrious** one at that.
Original (pos.)	Manages to be both hugely **entertaining** and uplifting.
Adversary (neg.)	Manages to be both hugely **funnier** and uplifting.
	Rotten Tomatoes Movie Reviews (MR) for CNN (Labels:**Positive, Negative**)
Original(pos.)	The film tunes into a grief that **could** lead a man across centuries.
Adversary(neg.)	The film tunes into a grief that **gotten** lead a man across centuries.
Original(neg.)	Bad in such a bizarre way that it's almost worth seeing, if only to witness the **crazy** confluence of purpose and taste.
Adversary(pos.)	Bad in such a bizarre way that it's almost worth seeing, if only to witness the **wack** confluence of purpose and taste.
	AG's News (AG) for CNN (Labels:**World, Sports, Business, Sci/Tech**)
Original(Business)	New york times co announces plan to sell manhattan building the new york times co plans to sell its building on west 43rd street in manhattan to a partnership led by tishman speyer **properties** , the **companies** announced monday.
Adversary (Sci/Tech)	New york times co announces plan to sell manhattan building the new york times co plans to sell its building on west 43rd street in manhattan to a partnership led by tishman speyer **functionality** , the **endeavour** announced monday.
Original (Sci/Tech)	**Group** questions e voting security black box voting hopes to halt the use of diebold 's voting machines.
Adversary(World)	**Factions** questions e voting security black box voting hopes to halt the use of diebold 's voting machines.
Original (Sci/Tech)	Coping with the common cold by karen pallarito, healthday reporter healthdaynews determined this cold season to nip your sneezing, runny nose and scratchy throat in the bud before those nasty respiratory symptoms **sideline** you? there 's a broad array of cold remedies you might want to try , ranging from over the counter preparations to basic ingredients tucked away in your kitchen pantry so what 'll it be? a combination pain reliever and nasal decongestant? vitamin c and echinacea? tea with honey? a brimming bowl of chicken soup? it turns out the best advice for dealing with the misery of a cold is the same principle mothers often apply when trying to coax their unruly toddlers to take a nap whatever works.
Adversary (Sports)	Coping with the common cold by karen pallarito , healthday reporter healthdaynews determined this cold season to nip your sneezing , runny nose and scratchy throat in the bud before those nasty respiratory symptoms **teammates** you? there 's a broad array of cold remedies you might want to try , ranging from over the counter preparations to basic ingredients tucked away in your kitchen pantry so what 'll it be? a combination pain reliever and nasal decongestant? vitamin c and echinacea? tea with honey? a brimming bowl of chicken soup? it turns out the best advice for dealing with the misery of a cold is the same principle mothers often apply when trying to coax their unruly toddlers to take a nap whatever works.
	SNLI for Bert (Labels:**Entailment, Neutral, Contradiction**)
Premise	A man in a yellow shirt and helmet mountain biking down a dusty path.
Original (Label: Neu.)	A man with headphones is **biking**.
Adversary (Label: Ent.)	A man with headphones is **motorcyclists**.
Premise	An older man in a light green shirt and dark green pants holds the hand of an older woman as they pass an outside eatery with teal umbrellas.
Original (Label: Ent.)	An **older** looking couple are holding hands as the walk past a restaurant outside.
Adversary (Label: Neu.)	An **elderly** looking couple are holding hands as the walk past a restaurant outside.

Table 3: Comparison of original and generated adversarial sentences from various datasets (MR, AG and SNLI) on BERT and CNN models. The blue font emphasizes the prediction labels and the red font emphasizes the perturbed words in the original and adversarial sentences.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Limitations are discussed in the section called Limitations following Section 7 (Conclusion).

☑ A2. Did you discuss any potential risks of your work?
The potential risks are also discussed in the section called Limitations.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
The paper's main claims are summarized in the abstract, introduction and conclusion part.

☑ A4. Have you used AI writing assistants when working on this paper?
Grammarly is used to check typos and grammar correctness.

B ☑ Did you use or create scientific artifacts?

Datasets, pre-trained models and representative baselines of adversarial attacks are used in the experimental part (Section 6).

☑ B1. Did you cite the creators of artifacts you used?
The creators of artefacts are cited and links are attached in Subsection 6.1 and Appendix B.1-B.3.

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
In Appendix B.1 (Dataset details), the number of sentences for training and testing is listed in Table 2.

C ☑ Did you run computational experiments?

The results of computational experiments are given in Subsections 6.2, 6.3, and Appendix B.4.

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
The computing infrastructure used is reported in Subsection 6.2 and the number of parameters in the models used is provided in Appendix B.1, B.3.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
The hyperparameters are introduced in Appendix B.3.

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
The average and standard deviation of bound computation is reported in Subsection 6.2 and Table 1. The computation runtime is shown via box plots in Subsection 6.2 and Figure 3. The convergence lines of the upper and lower bounds can be seen in Subsection 6.3 and Figure 4. The bound and estimation comparison under different subspace dimensions are shown in Appendix B.4 and Figure 6.

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
NLTK library is used for preprocessing in Subsection 4.1 and the links are attached in footnotes.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Not applicable. Left blank.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Not applicable. Left blank.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.