最好的防御就是进攻：修复文本对抗样本中的语义

Heng Yang[1],**Ke Li**[11]计算机科学系，埃克塞特大学，EX4 4QF，英国
埃克塞特 {hy345, k.li}@exeter.ac.uk

arXiv:2305.04067v2 [cs.CL] 1 Apr 2024

摘要

近期研究揭示了预训练语言模型易受对抗攻击影响的脆弱性。现有的对抗防御技术试图在特征空间或文本空间中重构对抗样本。然而，这些方法难以有效修复对抗样本中的语义，导致性能不尽如人意，并限制了其实际应用价值。为修复对抗样本中的语义，我们提出了一种名为反应式扰动散焦（RAPID）的新方法。RAPID 利用对抗检测器识别对抗样本的伪标签，并借助对抗攻击者修复对抗样本中的语义。在四个公开数据集上开展的大量实验，其实验结果令人信服地证明了 RAPID在各种对抗攻击场景下的有效性。针对以往研究中防御性能验证的问题，我们基于本研究提供了对抗检测与修复的演示，可通过以下链接轻松评估https://tinyurl.com/22ercuf8。

1 引言

预训练语言模型（PLM）已在各种自然语言处理任务中取得最先进水平（SOTA）的性能（Wang等人，2019年a,b）。然而，据报道，PLM极易受到对抗样本（又称对抗样本）的攻击（Li 等人，2019；Garg和Ramakrishnan，2020年；Li 等人，2020年；Jin 等人，2020年；Li 等人，2021年；Boucher等人，2022年）。这些对抗样本通过对自然样本（又称干净样本或良性样本）中的特定词语进行细微修改而生成（Morris 等人，2020）。尽管文本对抗鲁棒性对于应对对抗攻击的重要性已在深度学习社区中得到广泛认可（Alzantot等人，2018；Ren 等人，2019；Zang等人，2020；Zhang等人，2021；Jin 等人，2020年；Li 等人，2021年；Wang等人，2022年a；Xu等人，2023年），但提升对抗鲁棒性的工作仍十分有限，尤其是与计算机视觉等其他深度学习领域相比（Rony 等人，2019；Gowal等人，2021；Wang等人，2023年；Xu等人，2023年）。当前关于文本对抗鲁棒性的研究可分为三类——对抗防御、对抗训练（Liu 等人，2020年a,b；Ivgi和Berant，2021年；Dong等人，2021b,a）以及对抗样本重构（Zhou 等人，2019；Jones等人，2020年；Bao等人，2021；Keller等人，2021；Mozes 等人，2021；Li 等人，2022年；Shen等人，2023年）。由于对抗训练和重构都需要大量资源，人们对对抗防御的兴趣日益浓厚。然而，当前的对抗防御技术存在两个瓶颈。

现有研究几乎无法识别自然样本与对抗样本之间的语义差异[1]。以近期的一种对抗防御方法 RS&V（Wang等人，2022年b）为例。如图1所示，显然 RS&V 无法辨别对抗样本与修复后样本之间的语义差异。这是因为 RS&V 所采用的增强方法不仅没有针对性，也无法有效识别并消除对抗样本。

鉴于防御过程耗时较长，对抗防御也因计算效率低下而臭名昭著（Mozes 等人，2021年；Wang等人，2022年b）。这在一定程度上是因为它们无法预先检测对抗样本，因而对所有输入文本一概进行处理。这不仅将计算预算浪费在针对自然样本的不必要防御操作上，还会导致对自然样本采取不必要的防御立场，进而可能损害模型性能。

针对上述两项挑战，我们提出了一种简单而有效的文本对抗样本

[1]为简便起见，本文将对抗样本中的语义称为 PLM 编码的特征。

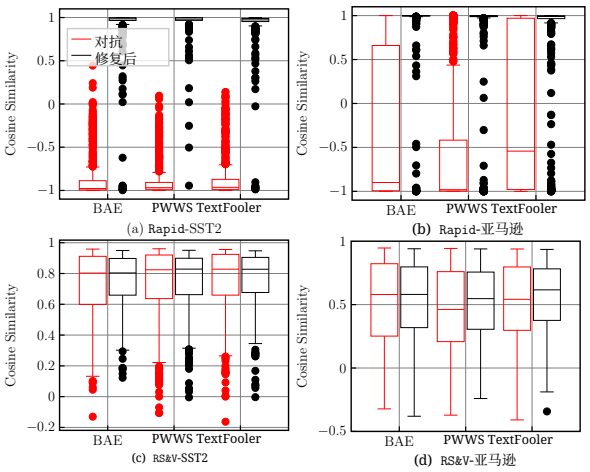


图1：RAPID 与 RS&V 所获得的对抗样本与自然样本对（以红色标出）以及修复后的对抗样本与自然样本对的余弦相似度箱线图。余弦相似度分别基于 RAPID 和 RS&V 的受害模型提取的特征进行评估。余弦相似度越大，相应的样本对就越相似。可以观察到，受害模型无法辨别 RS&V 生成的对抗样本与修复后的对抗样本之间的语义差异，而 RAPID 则能够准确区分对抗样本与自然样本。相反，使用 RAPID 时，修复后的对抗样本重新与自然样本在语义上保持一致。

防御范式，名为反应式扰动散焦（RAPID），具有以下两个鲜明特点。

- 为解决第一个瓶颈，我们提出了扰动去聚焦这一新概念（第2.2.2节）。其基本思想是利用对抗攻击者向预先检测出的对抗样本中重新注入一些扰动，以分散受害模型对恶意扰动的注意力，并基于受害模型的内在鲁棒性修复这些对抗样本。此外，我们通过伪语义相似度过滤策略（第2.2.3节）提高对抗防御的准确率。

- 为克服第二个瓶颈，RAPID训练一种位于受害模型内部的对抗检测器，而无需引入额外成本（第2.1节），从而主动将防御工作集中于预先检测为对抗样本的样本上。具体而言，该对抗检测器以多任务方式与受害模型联合训练，并能够识别由不同攻击者生成的对抗样本。这不仅有助于最大限度地减少对自然样本的附带影响（Xu等人，2022年），还减少了在防御自然样本上计算预算的浪费。

图2以情感分析为例，直观展示了 RAPID 的工作机制。我们的实证研究得出四项关键结论。

- RAPID 在预先检测到对抗样本后，可实现高达 99.9% 的修复准确率，显著优于文本／特征级重构和基于投票的方法（表2）。

- RAPID 在对抗防御中，较对抗攻击几乎降低 50% 的计算成本（表12）。

- RAPID 能够稳健地识别并防御多种未知的对抗攻击（表 4），例如 CLARE (Li 等人，2021年）以及ChatGPT-3.5(OpenAI，2023年）等大型语言模型。

我们开发了一个便于用户使用的 API[2]，作为基准测试平台，用于评测 RAPID 防御下的不同对抗攻击者。

2 提出的方法

我们提出的 RAPID 框架包含两个阶段。阶段 #1 训练一个联合模型，该模型不仅执行标准文本分类任务，还能够检测对抗样本。第2阶段基于 PD 实施伪监督对抗防御。该阶段将受害模型的注意力从恶意扰动上转移开，并在不影响自然样本性能的前提下修正输出。

2.1 阶段 #1：联合模型训练

阶段 #1 的核心是联合训练两个模型：一个是作为标准文本分类器的受害模型，另一个是受害模型内对抗检测器。后者是一个二分类器，可在防御前预先检测对抗样本。

2.1.1 基于多种攻击的对手采样

为了生成用于训练对抗检测器的数据，我们在受害模型上应用对抗攻击方法 F_S 来采样对抗样本。为了使对抗检测器能够识别各种未知的对抗样本，我们采用了三种广泛使用的开源对抗攻击器：BAE（Garg 和 Ramakrishnan，2020年）、PWWS（Ren 等人，2019年），

[2]为满足匿名要求，我们承诺在本文被接收后发布此工具。

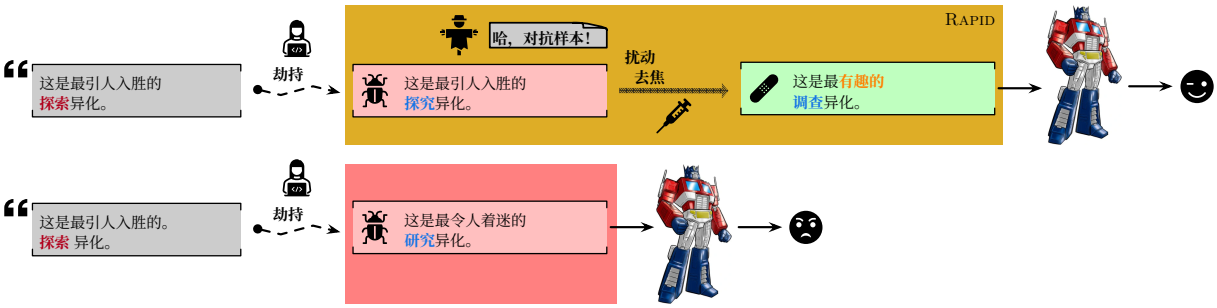


图 2：RAPID 在情感分析中的示例。该示例中的原词是 探索。扰动去聚焦通过注入扰动（有趣的）来分散目标模型对恶意扰动（即调查）的注意力，从而修复对抗样本。RAPID 仅对预先检测出的对抗样本实施防御。

与 TEXTFOOLER (Jin 等人, 2020年）。对于每个数据实例 $\langle \mathbf{x}\ y\rangle \in \mathcal{D}$，即自然样本集，我们将每种对抗攻击者应用于样本，生成三个对抗样本[3]:

$$\langle \tilde{\mathbf{x}}, \tilde{y}\rangle_i \leftarrow \mathcal{A}_i\left(F_S, \langle \mathbf{x}, y\rangle\right), \tag{1}$$

其中 $\mathcal{A}_i$、 $i \in \{1,2,3\}$分别表示 BAE、PWWS 和 TEXTFOOLER。 $\langle \tilde{\mathbf{x}}\ \tilde{y}\rangle_i$ 是由 $\mathcal{A}_i$生成的对抗样本。请注意，我们收集了所有对抗样本，包括成功和失败的样本，以构成对抗数据集 $\mathcal{D}$。最后，如图 3左侧所示，我们构建了一个混合数据集。 $\mathcal{D}:=\mathcal{D}\bigcup\mathcal{D}$ 用于联合模型训练。

2.1.2 联合模型训练目标

为了对受害模型和对抗检测器进行联合模型训练，我们提出如下聚合损失函数:

$$\mathcal{L} := \mathcal{L}_{\mathrm{c}} + \mathcal{L}_{\mathrm{d}} + \mathcal{L}_{\mathrm{a}} + \lambda||\boldsymbol{\theta}||_2^2, \tag{2}$$

其中， λ 是 ℓ_2 正则化参数， $\boldsymbol{\theta}$表示底层预训练语言模型（PLM）的参数。$\mathcal{L}_{\mathrm{c}}$、 $\mathcal{L}_{\mathrm{d}}$和 $\mathcal{L}_{\mathrm{a}}$ 分别表示训练标准分类器、对抗检测器和对抗训练所用的损失。

- 标准分类损失 $\mathcal{L}_{\mathrm{c}}$：此处采用文本分类中广泛使用的交叉熵损失:

$$\mathcal{L}_{\mathrm{c}} := -\sum_{i=1}^{C}\left[p_i \log\left(\tilde{p}_i\right) + q_i \log\left(\tilde{q}_i\right)\right], \tag{3}$$

其中， C 是类别数。 p 和 $\tilde{p}$分别表示标准分类标签的真实概率分布和预测概率分布，而 q 和$\hat{}q$ 表示任意的分别表示错误的标准分类标签及其可能性。请注意，在此损失中，$\mathcal{D}$ 中的对抗样本标签被设为虚拟值 $\varnothing$。这样，我们可以确保 $\mathcal{L}_{\mathrm{c}}$ 专注于自然样本。

- 对抗检测损失 $\mathcal{L}_{\mathrm{d}}$：该损失仅计算 $\mathcal{D}$中自然样本和对抗样本的二元交叉熵，实际使用的标签为 0 或 1。请注意，$\mathcal{L}_{\mathrm{d}}$ 用于将对抗检测器训练为二分类器，以判断输入样本是否为对抗样本。

- 对抗训练损失 $\mathcal{L}_{\mathrm{a}}$：实际中， $\mathcal{L}_{\mathrm{a}}$ 的计算方式与 $\mathcal{L}_{\mathrm{c}}$相同。为了提高对抗样本的鲁棒性，$\mathcal{L}_{\mathrm{a}}$ 仅计算对抗样本的损失，并将 $\mathcal{D}$中的自然样本标签设为虚拟 $\varnothing$。这样可以避免对抗训练损失对纯自然样本的性能产生负面影响；近期研究已指出，这种负面影响相当严重（Dong等人，2021a，b）。

总而言之，每个实例 $\langle \mathbf{x},\mathbf{y}\rangle^{\top}\in\mathcal{D}$ 都会附加三个不同的标签，以适配这三种训练损失，其中 $\mathbf{y}:=(y_1, y_2, y_3)^{\top}$。

2.2 第2阶段：响应式对抗防御

为了解决第1节讨论的效率和语义挑战，响应式对抗防御包括以下三个步骤。

2.2.1 对抗防御检测

我们的初步实验表明，BERT 和 DEBERTA 等预训练语言模型对对抗攻击造成的语义偏移十分敏感。因此，与当前通常对所有输入样本一概进行防御的对抗防御方法不同，我们将首先应用在阶段 #1 中训练的联合模型 F_J，通过以下预测来判断输入 $\hat{}\mathbf{x}$ 是否为对抗样本:

[3]词级对抗攻击的表述见附录A。

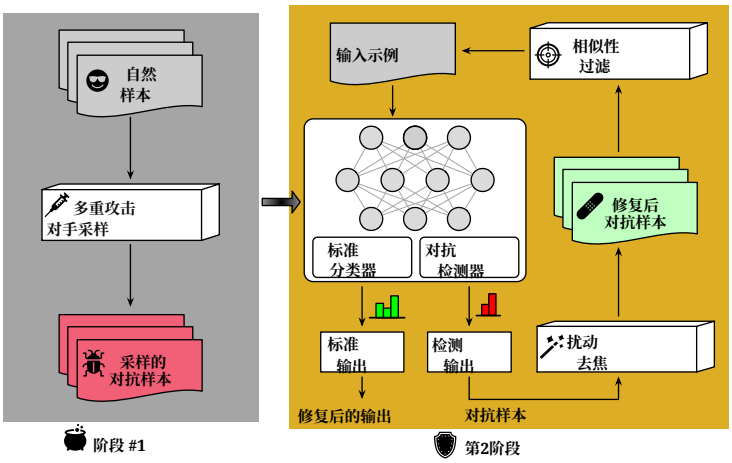


图3：RAPID 的总体架构与工作流程。

如下：

$$(\hat{y}_1, \hat{y}_2, \hat{y}_3) \leftarrow F_J(\hat{\mathbf{x}}), \tag{4}$$

其中 $\hat{}y_1$、$\hat{}y_2$和 $\hat{}y_3$ 分别是根据公式（2）中的三个训练损失所预测的标签。此后，仅将被识别为对抗样本的输入（即具有 $\hat{}y_2 = 1$的输入）用于后续的扰动去聚焦。

2.2.2 扰动去聚焦

这种扰动去聚焦的基本思想是，将 安全的 扰动注入第$\hat{}\mathbf{x}$ 中，该样本由第2.2.1节的对抗防御检测所识别。该过程如第2阶段的图3所示。在实践中，我们使用对抗攻击者对攻击 $\hat{}\mathbf{x}$，以获得一个修复后样本：

$$\langle \tilde{\mathbf{x}}^{\mathrm{r}}, \tilde{y}_1^{\mathrm{r}} \rangle \leftarrow \hat{\mathcal{A}}_{PD}\left(F_J, \langle \hat{\mathbf{x}}, \hat{y}_1 \rangle\right), \tag{5}$$

$\hat{}$ 其中，$\hat{}y_1$ 是 $\hat{}\mathbf{x}$的预测标签，$\mathcal{A}_{PD}$ 是一个对抗攻击者[4]。请注意，由于上述扰动不会改变 $\hat{}\mathbf{x}$的语义，因此被认为是安全的。通过这种方式，我们将标准分类器的注意力从恶意扰动上转移开，使标准分类器能够专注于对抗样本的原始语义。从本质上说，修复后样本可以根据其自身的鲁棒性得到正确分类。

2.2.3 伪语义相似度筛选

最后但同样重要的是，为防止修复后的对抗样本被误分类，我们提出了一种特征级伪语义相似度过滤策略，以减轻语义偏差。具体而言，对于每个 $\hat{}\mathbf{x}$，我们生成一组修复后样本 $\mathcal{S} := \{\tilde{}\mathbf{x}_i^{\mathrm{r}}\}_{i=1}^{k}$。然后，我们使用 F_J 对这些修复后样本进行编码，以提取其语义特征。随后，对于 $\mathcal{S}$ 中的每个修复后样本，我们计算其相似度分数，如下所示：

$$s_i = \frac{\sum_{j=1, j\neq i}^{k} \mathrm{sim}(\mathcal{H}_i, \mathcal{H}_j)}{k}, \tag{6}$$

其中，$\mathcal{H}_i$ 和 $\mathcal{H}_j$ 是由 F_J编码的 $\tilde{}\mathbf{x}_i^{\mathrm{r}}$ 和 $\tilde{}\mathbf{x}_j^{\mathrm{r}}$的隐藏状态，$\mathrm{sim}(*,*)$用于计算余弦相似度。为提高效率，本文将 $k = 3$设为该值。防御后，修复后$\hat{}\mathbf{x}$ 的标签被设为 $\mathcal{S}$ 中与修复后样本相似度得分最高的样本所预测的标签。

注释 1. 一般而言，对抗攻击者的基本思路是向自然样本中注入一些（通常有限的）恶意扰动，从而欺骗受害模型。这通常会使对抗样本看起来与自然样本相似。然而，扰动后相应的语义往往会遭到“破坏”。这启发我们引入一种新的对抗攻击者；尽管它不同于恶意攻击者，但在已知对抗样本伪标签的前提下，它能够攻击并修复对抗样本中被破坏的语义。此外，由于改变预测时遵循最小化编辑原则，我们还可以缓解修复后样本中的文本空间偏移，同时修复其语义。

备注 2。 请注意，RAPID 中的防御器与对抗检测器是解耦的，其性能与用于此对手采样的对抗攻击者无关。表4 中的实证结果表明，即使仅在少量对抗样本上进行训练，对抗检测器也能适应未知攻击方法。

[4]我们选择 PWWS，因为它成本效益高，且可由任意一种（或多种组合的）对抗攻击者替代。

3 实验设置

在本节中，我们将介绍实验所用的设置。

表 1：用于评估 RAPID 的数据集统计信息。由于资源占用较高，我们和先前的工作一样，使用亚马逊、AGNews 和 Yahoo! 数据集的子集来评估 RAPID。

数据集	类别	样本数量		
		训练	验证	测试
SST2	2	6,920	872	1,821
亚马逊	2	7,000	1,000	2,000
AGNews	4	120，000	0	7，600
Yahoo!	10	1，400，000	0	60，000

受害模型： 虽然任何预训练语言模型都可以以插件方式用于RAPID，本文考虑了BERT（Devlin 等人，2019年）和DEBERTA （He 等人，2021年），这两种基于 Transformer 结构、广泛使用的预训练语言模型[5]，，并将其同时用作受害分类器和联合模型。相应的超参数设置见附录B.2。

数据集： 我们考虑了三个广泛使用的文本分类数据集[6]，，包括SST2 （Socher 等人，2013）、亚马逊（Zhang 等人，2015）和AGNews （Zhang 等人，2015），其关键统计数据列于表1。SST2 和亚马逊是二元情感分类数据集。AGNews 和Yahoo！是多类别新闻分类数据集，分别包含 4 和10 个类别。

对抗攻击者： 我们的实验采用了TEXTAT-攻击[7]（Morris 等人，2020年）提供的三个开源攻击者。其功能概述如下，工作机制详见附录B.1。

a) 对手采样。使用 BAE、PWWS 和 TEXTFOOLER 对对抗样本进行采样，以训练对抗检测器（第 2.1 节）。由于它们代表不同类型的攻击，因此我们能够训练出一个可以识别多种对抗攻击的检测器。b) 对抗样本修复。我们在扰动去聚焦过程中使用 PWWS 作为攻击者 $\mathcal{A}_{PD}$ （第 2.2 节）。与 BAE 相比，我们的初步实验表明，PWWS 很少改变自然样本的语义，而且其计算效率高于 TEXTFOOLER。

c) 泛化能力评估。我们使用 IGA（Wang等人，2021a）、DEEPWORD- BUG（Gao等人，2018）、PSO（Zang等人，2020年）和 CLARE 来评估 RAPID 的泛化能力。

评估指标： 我们使用以下五项细粒度指标[8] 评估文本分类中的对抗防御性能。

- 自然准确率(NTA)：评估受害模型在仅包含自然样本的目标数据集上的性能。
- 攻击准确率(ATA)：评估受害模型在对抗攻击下的性能。
- 检测准确率(DTA)：衡量防御器检测对抗样本的性能。
- 防御准确率 (DFA)：评估防御器修复对抗样本的性能。
- 修复后准确率(RPA)：用于评估受攻击数据集经过修复后受害模型的性能。

请注意，我们在整个测试集上评估对抗检测和防御性能，而现有工作（Xu等人，2022年；Yang 等人，2022年；Dongetal.，2021a，b）仅评估从测试集中抽取的少量数据。

基线方法： 我们将 RAPID 与以下六种对抗防御基线方法进行比较。

- DISP（Zhou 等人，2019年）：这是一种嵌入特征重建方法。它使用扰动判别器评估某个词元受到扰动的概率，并提供一组潜在扰动。对于每个潜在扰动，嵌入估计器会基于上下文学习恢复原始词语的嵌入。
- FGWS（Mozes 等人，2021年）：该方法利用频率引导的词替换，借助对抗性词语替换的频率特征来检测对抗样本。
- RS&V（Wang等人，2022b）：这是一种基于随机替换投票策略的文本重构方法。RS&V通过随机将对抗样本中的词语替换为同义词，生成大量样本，并累积这些样本的逻辑值。

请注意，选择上述三种基线方法的理由在于它们均为开源，而对于 TEXTSHIELD (Shen 等人，2023年)等其他方法，我们几乎无法复现其实验结果。

[5] https://github.com/huggingface/transformers [6] 我们已在补充材料中发布详细的源代码和处理后的数据集。[7] https://github.com/QData/TextAttack

[8] 这些评估指标的数学定义见附录B.3。

4 实验结果

4.1 对抗样本检测性能

表 2 所示结果证明了 RAPID 中对抗检测器的有效性。该受害模型内对抗检测器与标准分类器联合训练，能够准确识别大多数数据集中的对抗样本。与以往基于对抗样本检测的防御方法（Mozes 等人，2021年；Wang等人，2022b；Shen 等人，2023年）相比，受害模型内对抗检测器无需额外成本即可识别对抗样本。另一方面，我们的评估证实，对自然样本进行对抗样本检测时，误报率极低（≈ 2%），因此对自然样本的性能影响微乎其微。此外，表 4 展示了 RAPID 对先前未见过的攻击方法的适应能力，凸显了我们对抗检测器的多功能性。它通过检测恶意攻击者引入的干扰（如语法错误和用词不当），出色地识别对抗样本。请注意，AGNews 数据集上的检测性能较低，这是因为 BERT 训练语料中没有新闻数据，He 等人（2021年）的表 8 对此已有讨论。

4.2 对抗样本防御性能

关于对抗防御，如表 2 所示，RAPID 在所有数据集上的表现均优于现有方法。针对正确识别出的对抗样本，RAPID 能够有效修复其中 92% 至 99% 的样本，即使面对具有挑战性的 10 类 Yahoo 数据集也是如此。我们的研究还揭示了无监督文本级和特征级重构方法的局限性，相关研究包括 Zhou 等人（2019年）；Mozes 等人（2021年）；Wang等人（2022b）。这些方法难以纠正对抗样本中的深层语义，因此效率低下且效果逊色。此外，我们发现，先前的方法在防御短文本中的对抗样本时不够稳健，在 SST2 和亚马逊数据集上的失效便证明了这一点。RAPID 始终能取得更高的防御准确率，尤其是在二分类数据集上。总之，RAPID 利用对抗攻击者修复对抗样本的深层语义，并尽量减少文本空间中的编辑，从而实现令人满意的对抗防御。我们强调，针对无监督特征和文本空间重构进行对抗防御时，专门修复深层语义至关重要。

4.3 消融实验

我们进行了消融实验，以评估伪语义相似度过滤（第 2.2.3 节）的有效性。该策略仅影响防御过程，因此我们省略了不受影响的指标，例如表 2 中的检测准确率。从表 3 所示结果可以看出，在大多数情况下，未采用该过滤策略的 RAPID，其对抗防御性能明显较差（ ≈ 1% ）。此外，与 SST2 和亚马逊数据集相比，在 AGNews 和 Yahoo! 数据集上，防御性能的下降更为显著。这种差异是由于 AGNews 和 Yahoo! 数据集的词汇量更大、文本更长，导致修复后的样本在相似度方面更加多样。

4.4 进一步的研究问题

我们通过回答以下研究问题（RQs），进一步探讨 RAPID 的相关发现。

研究问题1：RAPID 对未知攻击者的泛化能力如何?

方法：为了评估 RAPID 中受害模型内对抗检测器的泛化能力，我们针对多种最先进的对抗攻击者进行了实验：PSO、IGA、DEEP-WORDBUG 和 CLARE。这些攻击者均未参与 RAPID 中对抗检测器的训练。请注意，针对未知对抗攻击者取得更好的对抗检测和防御性能，意味着 RAPID 的泛化能力更强。

结果：从表4的结果中可以看出，在考量对抗检测性能时，RAPID 能够识别 SST2 和亚马逊数据集上多达 98.67% 的对抗样本。在对抗防御方面，RAPID 能够修复由各种未知攻击方法生成的大量对抗样本（在 SST2 和亚马逊数据集上分别高达 87.68% 和 94.65%）。然而，面对具有挑战性的 CLARE 攻击时，RAPID 识别和防御对抗样本的性能有所下降。这种性能下降可能是由于其对抗检测效果不佳所致；在 RAPID 中使用基于 CLARE 生成的对抗样本进行对抗检测训练，或许能够改善这一情况。总之，RAPID 展现出了稳健的泛化能力，能够有效检测并修复由未知攻击者生成的各类对抗样本。

研究问题2：扰动去聚焦是否真的会重新

表2：RAPID 在四个公开数据集上的主要对抗检测和防御性能。受害模型为 BERT，**粗体** 字体表示最佳性能。我们报告了五次随机运行的平均准确率。以往研究中报告的对抗防御性能因对抗攻击者的实现方式而异。为确保公平比较，所有基线实验均基于 Textattack 库中最新的对抗攻击者重新实现，以避免产生偏差。“TF”表示 TEXTFOOLER。

防御器 攻击者		AGNews（4 类）	Yahoo!（10 类）	SST2（2 类别）	亚马逊（2 类）
		NTA 对抗训练 DTA DFA RPA	NTA 对抗训练 DTA DTA RPA	NTA 对抗训练 DTA DFA RPA	NTA 对抗训练 DTA DFA RPA
DISP	PWWS	32.09 55.49 57.82 68.23	5.70 61。67 54。95 50。24	23.44 38.93 34.46 35.33	15。56 41。90 45。92 59。80
	TF	94.13 50.50 53.78 56.18 70.16	75.63 13.60 50.73 57.48 53.18	91。24 16。21 37。80 34。37 37。16	93.67 21.77 43.10 47.15 60.56
	BAE	74。80 45。26 45。75 81。39	27.50 54.82 53.75 50.90	35.21 36.59 37.51 42.22	44.00 40.28 42.74 61.85
FGWS	PWWS	32.09 65.24 68.35 71.78	5.70 65。83 61。46 53。28	23。44 40。28 40。38 39。20	15。56 44。47 56。89 60。29
	TF	94.25 50.50 68.88 70.71 73.40	76.24 13.60 68.57 65.17 54.53	91.34 16.21 42.79 41.05 41.53	94。26 21。77 45。75 58。74 61。51
	BAE	74.80 44.29 47.95 83.57	27.50 58.63 56.33 52.94	35.21 43.83 48.37 44.90	44.00 42.26 43.04 64.63
RS&V	PWWS	32.09 83.67 84.96 83.80	5.70 65.01 65.22 57.22	23.44 36.90 37.10 38.54	15.56 29.60 45.30 46.17
	TF	94。14 50。50 82。44 83。45 82。53	76。39 13。60 74。21 74。54 58。10	91.55 16.21 39.70 38.40 39.70	94.32 21.77 40.70 42.30 55.70
	BAE	74。80 46。98 48。67 86。90	27。50 37。41 37。88 62。27	35.21 19.84 20.92 43.65	44.00 38.59 39.01 65.03
RAPID	PWWS	32.09 **90.11 95.88 92.36**	5.70 **87.33 92.47 69.40**	23.44 **94.03 98.62 89.85**	15.56 **97.33 99.99 94.42**
	TF	94.30 50.50 **90.29 96.76 92.14**	76。45 13。60 **87。49 93。54 70。50**	91。70 16。21 **94。03 99。86 89。72**	94。24 21。77 **93。85 99。99 93。96**
	BAE	74.80 **57.55 96.25 93.64**	27.50 **82.46 96.30 73.06**	35.21 **78.99 99.28 89.77**	<escape>44.00 **80.55 99.99 93.89**</escape>

表3：RAPID **未采用伪相似度过滤**时的性能（彩色数字表示消融实验中的性能下降）。未受伪相似度过滤影响的指标已省略。

数据集	攻击者	DTA	RPA
AGNews	PWWS	94.19(−1.69 ↓)	90。80(−1.56 ↓)
	TF	94.26(−2.50 ↓)	91。35(−0.79 ↓)
	BAE	92.98(−3.27 ↓)	91.44(−2.20 ↓)
Yahoo!	PWWS	88。04(−4.43 ↓)	65.38(−4.02 ↓)
	TF	91。28(−2.26 ↓)	67。48(−3.02 ↓)
	BAE	92.48(−3.84 ↓)	71.35(−1.71 ↓)
SST2	PWWS	98.12(−0.50 ↓)	87。80(−2.05 ↓)
	TF	98。03(−1.83 ↓)	88。40(−1.32 ↓)
	BAE	95.87(−3.41 ↓)	87.52(−2.25 ↓)
亚马逊	PWWS	99。99（0。00）	94。40(−0.02 ↓)
	TF	98.92(−1.07 ↓)	93。31(−0.65 ↓)
	BAE	98.53(−1.41 ↓)	93。62(−0.27 ↓)

表4：RAPID 在对抗检测和防御方面的性能，**针对未知的对抗攻击**。

数据集	攻击者	ATA	DTA	DFA	RPA
AGNews	PSO	14.83	68.46	67.82	90.39
	IGA	26.87	76.74	74.59	92.33
	DEEPWORDBUG	45.53	72.73	87.23	89.33
	CLARE	8.46	62.78	61.54	64.78
Yahoo!	PSO	6.28	80.26	76.89	87.82
	IGA	14.75	82.69	81.02	54.55
	DEEPWORDBUG	51.34	72.73	87.10	62.27
	CLARE	3.56	64.85	62.40	52.47
SST2	PSO	7.95	87.50	87.50	82.61
	IGA	18.39	89.33	98.67	87.68
	DEEPWORDBUG	30.67	95.44	83.59	81.90
	CLARE	2.59	62.50	59.37	65.30
亚马逊	PSO	5.76	90.48	90.48	91.55
	IGA	14.91	92.31	92.31	94.65
	DEEPWORDBUG	43.43	87.04	85.19	86.87
	CLARE	3.25	60.44	59.37	62.94

成对对抗样本?

方法：为回答这一研究问题，我们考察对抗样本与其修复后对应样本在特征空间中的差异。具体而言，我们使用三种攻击方法（即 BAE、PWWS、TEXTFOOLER）生成对抗样本及其对应的修复后样本，并随机选取 1,000 个自然样本。我们使用受害模型将这些样本编码到特征空间中，并评估对抗样本—自然样本对与修复后对抗样本—自然样本对之间的余弦相似度。余弦相似度得分越高，表明修复对抗样本深层语义的效果越好。结果：图 1和4中的箱线图展示了通过成对语义相似度评估得到的相似度得分分布。这些图中的语义相似度得分分布（例如，修复后样本的中位相似度得分始终高于对抗样本）表明

RAPID 修复后的样本与自然样本之间存在显著的整体相似性，这意味着 RAPID 确实修复了对抗样本的深层语义。相反，使用 RS&V 获得的修复后样本，其相似度分数在所有数据集上都与对抗样本的相似度分数无法区分。这种情况在许多现有的对抗防御方法中都很常见。总之，我们的观察结果表明，RAPID 能够有效修复对抗样本的深层语义。

研究问题3：受害模型的内在鲁棒性如何影响 RAPID?

方法：我们评估了受害模型内在鲁棒性的影响，重点关注 DE- BERTA——一种用于各种任务的先进预训练语言模型。具体而言，我们基于 DEBERTA 训练了一个受害模型，复现了实验设置，并评估了性能变化

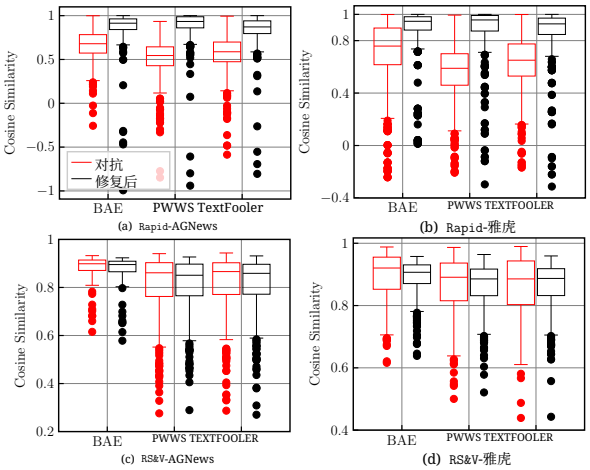


图4：多类别数据集上的语义余弦相似度分数分布箱线图。与图1类似，根据特征相似度分数分布，RAPID 更擅长修复语义。

表5：RAPID 在四个公开数据集上的表现**基于受害模型 DEBERTA**。红色表示与基于 BERT 的 RAPID 相比性能有所下降。

数据集 攻击者		NTA	ATA	DTA	DFA	RPA
AGNews	PWWS	96.69	62.77	96.47	98.47	93.12
	TF		39.85	91.41	95.90 ↓	93.69
	BAE		81.64	90.20	97.92	93.40 ↓
Yahoo!	PWWS	78.63	15.70	88.91	92.64	70.47
	TF		6.19	89.32	92.60	69.96 ↓
	BAE		47.50	90.25	93.74 ↓	72.12 ↓
SST2	PWWS	95.01	37.14	95.21	98.42	94.15
	TF		22.59	93.06 ↓	99.08	94.58
	BAE		38.84	80.82	98.59	94.16
亚马逊	PWWS	95.51	22.72	97.62	99.99	94.55
	TF		23.95	94.91	99.99	94.84
	BAE		56.65	82.71	99.99	94.50

RAPID 基于此 DEBERTA 受害模型。

结果：如表 5所示，基于 DEBERTA 的受害模型在对抗攻击下展现出更高的准确率，表明 DEBERTA 相较于基于 BERT 构建的受害模型具有更强的内在鲁棒性。具体而言，基于 DEBERTA的RAPID在所有分类数据集上识别对抗样本的表现均十分出色，尤其是在二分类数据集上。对抗检测和防御的性能也呈现出类似的上升趋势。这凸显了受害模型的鲁棒性对我们的方法有着重大影响，尤其是在增强对抗检测和防御方面。

5 相关工作

以往关于对抗防御的研究可分为三类：基于对抗训练的方法（Miyato等人，2017年；Zhu等人，2020；Ivgi和Berant，2021年）；基于上下文重构的方法（Pruthi等人，2019年；Liu等人，2020b；Mozes 等人，2021年；Keller等人，2021年；Chen等人，2021年；Xu等人，2022年；Li 等人，2022年；Swenor和Kalita，2022年）；以及基于特征重构的方法（Zhou 等人，2019年；Jones等人，2020；Wang等人，2021a）。一些研究（Wang等人，2021b）也探讨了混合防御方法。至于基于对抗训练的方法，它们因自然样本性能下降而声名狼藉。此类方法可以通过微调提升预训练语言模型的鲁棒性，但灾难性遗忘会增加模型训练成本（Dong等人，2021b）。基于文本重构的方法，例如词语替换（Mozes 等人，2021年；Bao等人，2021年）和基于翻译的重构，可能无法识别经过语义修复的对抗样本，或引入新的恶意扰动（Swenor和Kalita，2022年）。另一方面，基于特征重构的方法可能难以修复拼写错误攻击（Liu等人，2020a；Tan等人，2020；Jones等人，2020）、句子级攻击（Zhao等人，2018；Cheng等人，2019）以及其他未知攻击。一些研究致力于对抗检测与防御联合任务（Zhou 等人，2019年；Mozes 等人，2021年；Wang等人，2022b）。然而，这些对抗检测方法可能无法有效应对未知的对抗攻击者，也难以缓解对抗防御中的资源浪费。另一项与RAPID类似的工作是Textshield（Shen 等人，2023年），它旨在通过基于显著性的检测器检测对抗句子，并使用纠正器修复对抗样本，从而防御词级对抗攻击。总体而言，我们的研究着重于向对抗样本引入最小安全扰动，以保持语义，从而缓解所有基于重构的方法中的语义偏移问题。

6 结论

我们提出了一种新颖的对抗防御方法，即扰动去聚焦，用于修复对抗样本中的语义。RAPID 解决了先前研究中的语义偏移问题。RAPID 在修复对抗样本方面表现出色（正确识别的对抗样本比例最高达 $\approx 99\%$）。我们认为，扰动去聚焦有望显著改变文本对抗防御的格局。

局限性

所提方法的一个局限是，它倾向于向对抗样本中引入新的扰动，这可能导致语义偏移。这对于某些任务（例如机器翻译）而言可能并不安全。此外，该方法在训练阶段需要大量计算资源来生成对抗样本，这在某些场景中可能构成限制。最后，该方法尚未在广泛的自然语言处理任务和领域上进行测试，因此有必要在其他任务和领域开展进一步评估，以全面衡量其能力。

参考文献

Moustafa Alzantot、Yash Sharma、Ahmed Elgohary、Bo- Jhang Ho、Mani B. Srivastava 和 Kai-Wei Chang。2018。生成自然语言对抗样例。载于 *EMNLP'18*：*2018*年自然语言处理实证方法会议论文集，第 2890–2896 页。计算语言学协会。

Rongzhou Bao、Jiayi Wang 和 Hai Zhao。2021年。在不牺牲性能的情况下防御预训练语言模型免受对抗性词语替换攻击。载于*ACL-IJCNLP'21*：*2021*年计算语言学协会会议论文集：*ACL-IJCNLP 2021*，线上活动，*2021*年*8*月*1*日至*6*日，ACL-IJCNLP 2021 卷，收录于 *ACL*论文集，第 3248–3258 页。计算语言学协会。

Nicholas Boucher、Ilia Shumailov、Ross Anderson 和 Nicolas Papernot。2022年。恶意字符：难以察觉的自然语言处理攻击。载于第*43*届 *IEEE* 安全与隐私研讨会，*SP 2022*，旧金山，加利福尼亚州，美国，*2022*年*5*月*22*日至*26*日，第1987–2004页。IEEE。

Guandan Chen、Kai Fan、Kaibo Zhang、Boxing Chen 和 Zhongqiang Huang。2021年。用于神经机器翻译的流形对抗增强。载于 *ACL-IJCNLP'21*：*2021*年计算语言学协会会议论文集，第3184–3189页。计算语言学协会。

Yong Cheng、Lu Jiang 和 Wolfgang Macherey。2019年。采用双重对抗输入的鲁棒神经机器翻译。载于*ACL'19*：第*57*届计算语言学协会会议论文集，第4324–4333页。计算语言学协会。

Jacob Devlin、Ming-Wei Chang、Kenton Lee 和 Kristina Toutanova。2019年。BERT：用于语言理解的深度双向变换器预训练。载于*NAACL-HLT'19*：*2019*年论文集北美计算语言学协会分会年会，第 4171–4186 页。计算语言学协会。

董新帅、Anh Tuan Luu、纪荣嵘和刘宏。2021a。提升对自然语言词语替换的鲁棒性。收录于*ICLR'21*：第九届国际学习表征会议论文集。OpenReview.net。

董新帅、Anh Tuan Luu、Min Lin、Shuicheng Yan 和张汉旺。2021b。预训练语言模型应如何微调以增强对抗鲁棒性？收录于*NeurIPS'21*：*2021*年神经信息处理系统大会论文集，第 4356–4369 页。

Javid Ebrahimi、Anyi Rao、Daniel Lowd 和窦德景。2018。Hotflip：用于文本分类的白盒对抗样本。收录于第 *56* 届计算语言学协会年会论文集，*ACL 2018*，澳大利亚墨尔本，*2018*年*7*月*15*日至*20*日，第 *2* 卷：短论文，第 31–36 页。计算语言学协会。

高吉、Jack Lanchantin、Mary Lou Soffa 和 Yanjun Qi。2018年。用于逃避深度学习分类器的对抗文本序列黑盒生成。载于*SP'18*：*2018*年*IEEE*安全与隐私研讨会论文集，第50–56页。IEEE计算机学会。

Siddhant Garg 和 Goutham Ramakrishnan。2020年。BAE：基于BERT的文本分类对抗样本。载于*EMNLP'20*：*2020*年自然语言处理经验方法会议论文集，第6174–6181页。计算语言学协会。

Sven Gowal、Sylvestre-Alvise Rebuffi、Olivia Wiles、Florian Stimberg、Dan Andrei Calian 和 Timothy A. Mann。2021年。使用生成数据提升鲁棒性。载于*NeurIPS'21*：神经信息处理系统进展，第4218–4233页。

Chuan Guo、Alexandre Sablayrolles、Hervé Jégou 和 Douwe Kiela。2021年。针对文本变换器的基于梯度的对抗攻击。载于*2021*年自然语言处理实证方法会议论文集（*EMNLP 2021*），线上活动／蓬塔卡纳，多米尼加共和国，*2021*年*11*月*7*日至*11*日，第5747–5757页。计算语言学协会。

何鹏程、高剑峰和陈卫竹。2021年。DeBERTaV3：通过梯度解耦的嵌入共享，采用 ELECTRA 风格的预训练来改进 DeBERTa。*CoRR*，abs/2111.09543。

Maor Ivgi 和 Jonathan Berant。2021年。通过离散对抗训练实现模型鲁棒性。载于*EMNLP'21*：*2021*年自然语言处理实证方法会议论文集，第 1529–1544 页。计算语言学协会。

Di Jin、Zhijing Jin、Joey Tianyi Zhou 和 Peter Szolovits。2020年。BERT 真的鲁棒吗？文本分类与蕴涵任务中自然语言攻击的强基线。载于*AAAI'20*：第*34*届 *AAAI* 人工智能大会论文集，第8018–8025页。AAAI 出版社。

Erik Jones、Robin Jia、Aditi Raghunathan 和 Percy Liang。2020年。稳健编码：应对对抗性拼写错误的框架。载于*ACL'20*：计算语言学协会第*58*届年会论文集，第2752–2765页。计算语言学协会。

Yannik Keller、Jan Mackensen 和 Steffen Eger。2021年。BERT 防御：一种基于 BERT、用于应对认知启发式正字法对抗攻击的概率模型。载于*ACL-IJCNLP'21*：*2021*年计算语言学协会会议论文集，收录于*ACL*论文集第 ACL-IJCNLP 2021 卷，第1616–1629页。计算语言学协会。

Dianqi Li、Yizhe Zhang、Hao Peng、Liqun Chen、Chris Brockett、Ming-Ting Sun 和BillDolan。2021年。用于文本对抗攻击的上下文化扰动。载于*NAACL-HLT'21*：北美计算语言学协会分会*2021*年会议论文集，第5053–5069页。计算语言学协会。

Jinfeng Li、Shouling Ji、Tianyu Du、Bo Li 和 Ting Wang。2019年。Textbugger：针对现实应用生成对抗文本。载于第 *26* 届年度网络与分布式系统安全研讨会（*NDSS 2019*），*2019* 年 *2* 月 *24—27* 日，美国加利福尼亚州圣迭戈。互联网协会。

Linyang Li、Ruotian Ma、Qipeng Guo、Xiangyang Xue 和 Xipeng Qiu。2020年。BERT-ATTACK：使用 BERT 对 BERT 发起对抗攻击。载于*EMNLP'20*：*2020* 年自然语言处理实证方法会议论文集，第 6193–6202 页。计算语言学协会。

Linyang Li、Demin Song、Jiehang Zeng、Ruotian Ma 和 Xipeng Qiu。2022年。重建与集成：探索针对文本对抗样本的防御。*CoRR*，abs/2203.14207。

Hui Liu、Yongzheng Zhang、Yipeng Wang、Zheng Lin 和 Yige Chen。2020年a。联合字符级词嵌入与对抗稳定性训练，以防御对抗文本。载于*AAAI'20*：第*34*届 *AAAI* 人工智能大会论文集，第 8384–8391 页。AAAI 出版社。

Kai Liu、Xin Liu、An Yang、Jing Liu、Jinsong Su、Sujian Li 和 Qiaoqiao She。2020b。一种用于机器阅读理解的鲁棒对抗训练方法。发表于*AAAI'20*：第三十四届 *AAAI* 人工智能大会论文集，第 8392–8400 页。AAAI 出版社。

Takeru Miyato、Andrew M. Dai 和 Ian J. Goodfel- low。2017年。半监督文本分类的对抗训练方法。载于*ICLR'17*：第五届国际表征学习会议论文集。OpenReview.net。

John X. Morris、Eli Lifland、Jin Yong Yoo、Jake Grigsby、Di Jin 和 Yanjun Qi。2020年。TextAttack：一个用于自然语言处理中对抗攻击、数据增强和对抗训练的框架。载于*2020*年自然语言处理实证方法会议论文集：系统演示，*EMNLP 2020* - 演示，线上，*2020*年*11*月*16*日至*20*日，第 119–126 页。计算语言学协会。

Maximilian Mozes、Pontus Stenetorp、Bennett Kleinberg 和 Lewis D. Griffin。2021年。用于检测文本对抗样本的频率引导词替换。载于*EACL'21*：第*16*届欧洲计算语言学协会分会会议论文集，第 171–186 页。计算语言学协会。

OpenAI。2023年。GPT-4技术报告。*CoRR*，abs/2303.08774。

Danish Pruthi、Bhuwan Dhingra 和 Zachary C. Lipton。2019年。通过鲁棒的词语识别应对对抗性拼写错误。载于 *ACL'19*：计算语言学协会第*57*届会议论文集，第5582–5591页。计算语言学协会。

Shuhuai Ren、Yihe Deng、Kun He 和 Wanxiang Che。2019年。通过概率加权的词语显著性生成自然语言对抗样例。载于*ACL'19*：计算语言学协会第*57*届会议论文集，第1085–1097页。计算语言学协会。

Jérôme Rony、Luiz G. Hafemann、Luiz S. Oliveira、Ismail Ben Ayed、Robert Sabourin 和 Eric Granger。2019年。解耦方向与范数，以实现高效的基于梯度的L2对抗攻击和防御。载于 *CVPR'19*：*IEEE*计算机视觉与模式识别会议，第4322–4330页。计算机视觉基金会 / IEEE。

Lingfeng Shen、Ze Zhang、Haiyun Jiang 和 Ying Chen。2023年。TEXTSHIELD：超越在文本分类中成功检测对抗性句子。载于*ICLR'23*：第十一届国际学习表征会议。OpenReview.net。

Richard Socher、Alex Perelygin、Jean Wu、Jason Chuang、Christopher D. Manning、Andrew Y. Ng 和 Christopher Potts。2013年。基于情感树库的语义组合性递归深度模型。载于《*2013*年自然语言处理实证方法会议论文集》（*EMNLP 2013*），*2013*年*10*月*18*日至*21*日，美国华盛顿州西雅图市凯悦酒店，*SIGDAT*（*ACL* 的一个特别兴趣小组）会议，第1631–1642页。ACL。

Abigail Swenor 和 Jugal Kalita。2022年。使用随机扰动缓解情感分析模型中的对抗攻击。*CoRR*，abs/2202.05758。

Samson Tan、Shafiq R. Joty、Lav R. Varshney 和 Min- Yen Kan。2020 年。注意屈折变化！通过词基 - 屈折编码改进非标准英语的自然语言处理。载于 *EMNLP'20*：*2020*年自然语言处理中的经验方法会议论文集，第5647–5663页。计算语言学协会。

Alex Wang、Yada Pruksachatkun、Nikita Nangia、Aman- preet Singh、Julian Michael、Felix Hill、Omer Levy 和 Samuel R. Bowman。2019年a。Superglue：面向通用语言理解系统的更具挑战性的基准。载于*NeurIPS'19*：神经信息处理系统进展，第 3261–3275 页。

Alex Wang、Amanpreet Singh、Julian Michael、Fe- lix Hill、Omer Levy 和 Samuel R. Bowman。2019年b。GLUE：用于自然语言理解的多任务基准与分析平台。载于*ICLR'19*：第七届国际学习表征会议。OpenReview.net。

Boxin Wang、Hengzhi Pei、Boyuan Pan、Qian Chen、Shuohang Wang 和 Bo Li。2020年。T3：用于定向攻击的树自编码器约束对抗文本生成。载于*2020*年自然语言处理中的经验方法会议论文集，*EMNLP 2020*，线上，*2020*年*11*月*16*日至*20*日，第6134–6150页。计算语言学协会。

Boxin Wang、Chejian Xu、Xiangyu Liu、Yu Cheng 和 Bo Li。2022a。Semattack：通过不同语义空间进行自然文本攻击。载于计算语言学协会：*NAACL 2022* 论文集，*2022*年*7*月*10*日至*15*日，美国华盛顿州西雅图，第176–205页。计算语言学协会。

Xiaosen Wang、Jin Hao、Yichen Yang 和 Kun He。2021a。通过同义词编码实现自然语言对抗防御。载于 *UAI'21*：第*37*届人工智能不确定性会议论文集，第161卷，机器学习研究论文集，第823–833页。AUAI出版社。

Xiaosen Wang、Yifeng Xiong 和 Kun He。2022b。通过随机替换和投票检测文本对抗样本。载于 *UAI*，机器学习研究论文集第 180 卷，第 2056–2065 页。PMLR。

Xiaosen Wang、Yichen Yang、Yihe Deng 和 Kun He。2021b。针对基于同义词替换的文本攻击，采用快速梯度投影方法进行对抗训练。载于*AAAI'21*：第 *35* 届 *AAAI* 人工智能大会论文集，第 13997–14005 页。AAAI 出版社。

Zekai Wang、Tianyu Pang、Chao Du、Min Lin、Wei- wei Liu 和 ShuichengYan。2023年。更好的扩散模型进一步提升了对抗训练。*CoRR*，abs/2302.04638。

Jianhan Xu、Cenyuan Zhang、Xiaoqing Zheng、Linyang Li、Cho-Jui Hsieh、Kai-Wei Chang 和 Xuanjing Huang。2022年。通过学习重新加权干净样本，构建具有对抗鲁棒性的文本分类器。载于*ACL'22*：*2022*年计算语言学协会年会论文集（*Findings*），第1694–1707 页。计算语言学协会。

Yuancheng Xu、Yanchao Sun、Micah Goldblum、Tom Goldstein 和 Furong Huang。2023年。探索并利用对抗鲁棒性的决策边界动态。*CoRR*，abs/2302.03015。

Puyudi Yang、Jianbo Chen、Cho-Jui Hsieh、Jane-Ling Wang 和 Michael I. Jordan。2020年。贪心攻击与 Gumbel 攻击：针对离散数据生成对抗样本。机器学习研究杂志，21:43:1– 43:36。

Yichen Yang、Xiaosen Wang 和 Kun He。2022年。抵御词级对抗攻击的鲁棒文本嵌入。载于*UAI*，机器学习研究论文集第180卷，第2214–2224页。PMLR。

Yuan Zang、Fanchao Qi、Chenghao Yang、Zhiyuan Liu、Meng Zhang、Qun Liu 和 Maosong Sun。2020年。将词级文本对抗攻击视为组合优化。载于*ACL'20*：第*58*届计算语言学协会年会论文集，第6066–6080页。计算语言学协会。

Guoyang Zeng、Fanchao Qi、Qianrui Zhou、Tingji Zhang、Zixian Ma、Bairu Hou、Yuan Zang、Zhiyuan Liu 和 Maosong Sun。2021年。Openattack：一个开源文本对抗攻击工具包。载于第*59*届计算语言学协会年会与第*11*届国际计算语言学联合会议论文集，*ACL 2021*——系统演示，线上，*2021*年*8*月*1*日至*6*日，第363–371页。计算语言学协会。

Xiang Zhang、Junbo Jake Zhao 和 Yann LeCun。2015。用于文本分类的字符级卷积网络。载于神经信息处理系统进展第*28*卷：*2015*年神经信息处理系统年会，*2015*年*12*月*7*日至*12*日，加拿大魁北克省蒙特利尔，第649–657页。

Xinze Zhang、Junzhe Zhang、Zhenhua C hen 和 Kun He。2021年。为神经机器翻译构造对抗样本。载于*ACL-IJCNLP'21*：第*59*届计算语言学协会年会暨第*11*届自然语言处理国际联合会议论文集，第1967–1977页。计算语言学协会。

Zhengli Zhao、Dheeru Dua 和 Sameer Singh。2018年。生成自然对抗样本。载于*ICLR'18*：第六届国际学习表征会议论文集。OpenReview.net。

Yichao Zhou、Jyun-Yu Jiang、Kai-Wei Chang 和 Wei Wang。2019年。学习识别扰动，以阻止文本分类中的对抗攻击。载于 *EMNLP-IJCNLP'19*：自然语言处理经验方法会议暨第九届自然语言处理国际联合会议论文集，第4903–4912页。计算语言学协会。

Chen Zhu、Yu Cheng、Zhe Gan、Siqi Sun、Tom Gold- stein 和 Jingjing Liu。2020年。Freelb：面向自然语言理解的增强型对抗训练。载于*ICLR'20*：第八届国际学习表征会议论文集。OpenReview.net。

7 可复现性

为鼓励所有对我们的工作感兴趣的人实现RAPID，我们采取了以下措施：

- 我们创建了一个线上点击运行演示，可通过以下网址访问https://tinyurl.com/22ercuf8，以便轻松进行评估。任何人都可以输入对抗样本，并立即获得修复后的样本。

- 我们已发布详细的源代码和处理后的数据集，可在补充材料中获取。这使所有人都能使用官方实现，有助于理解论文，并便于开展自己的实现。

- 我们还将发布一个线上基准测试工具，用于评估 RAPID 防御下对抗攻击者的表现。这一举措对于减少不同代码库之间的评估差异至关重要。

这些工作旨在促进我们研究成果的可复现性，并便于研究社区加以实现。

A 对抗攻击

A.1 词级对抗攻击

我们的重点是防御词级对抗攻击。不过，我们的方法可以轻松适配不同类型的对抗攻击。设 $x = (x_1, x_2, \cdots, x_n)$ 为一个自然句子，其中 x_i、$1 \leq i \leq n$ 表示一个词。y 是真实标签。词级攻击者通常会用相似的词语（例如同义词）替换一些原始词语，以欺骗目标模型。例如，将 x_i 替换为 $\hat{x}_i$ 可生成一个对抗样本：$\hat{x} = (x_1, \cdots, \hat{x}_i, \cdots, x_n)$，其中 $\hat{x}_i$ 是一个用于替换 x_i 的替代替换。对于对抗样本 $\hat{x}$，目标模型 F 对其标签的预测如下：

$$\hat{y} = \operatorname{argmax} F\left(\cdot|\hat{x}\right), \tag{7}$$

其中，若 $\hat{y} \neq y$，则 $\hat{x}$ 是一个成功的对抗样本。为了表示使用对抗攻击者 $\mathcal{A}$ 对 F 发起的对抗攻击，我们将对抗攻击表示如下：

$$(\hat{x}, \hat{y}) \leftarrow \mathcal{A}(F, (x, y)), \tag{8}$$

其中，x 和 y 分别表示自然样本及其真实标签。$\hat{x}$ 和 $\hat{y}$ 分别表示扰动后的对抗样本和标签。

A.2 文本对抗攻击的研究

本节将深入考察文本对抗攻击。

传统方法，例如Li 等人 (2019年) 和 Ebrahimi 等人 (2018)，通常会对词语进行字符级修改（例如，将“good”改为“go0d”），通过改变其统计模式来欺骗模型。另一种方法是基于知识的扰动，以Zang 等人 (2020年) 的研究为例，这类方法利用 HowNet 等资源限制搜索空间，尤其是在词语替换方面。

近期研究（Garg 和 Ramakrishnan，2020年；Li 等人，2020年）探讨了使用预训练模型生成感知上下文的扰动（Li 等人，2021年）。基于语义的方法，例如 SemAttack（Wang等，2022a），通常利用 BERT 嵌入聚类来生成复杂的对抗样本。这不同于以往的启发式方法，后者采用贪心算法（Yang 等人，2020年；Jin 等人，2020年）或遗传算法（Alzantot 等人，2018；Zang 等人，2020年），以及侧重于句法限制的基于梯度的技术（Wang等，2020年；Guo 等人，2021年）。

随着对抗攻击技术的发展，Morris 等人，2020年）开发了 TextAttack（Zeng 等人，2021年）等众多工具，并将其开源发布。这些资源使深度学习研究人员能够以极少的代码高效评估对抗鲁棒性。因此，我们使用 TextAttack 框架开展对抗防御实验，并对 TextAttack 的作者和贡献者所付出的巨大努力表示感谢。

B 实验实现

B.1 实验对抗攻击器

我们采用 BAE、PWWS 和 TEXTFOOLER 生成对抗样本，用于训练对抗检测器。选择这些攻击者是因为它们代表了不同类型的攻击，使我们能够训练出一个可以识别多种对抗攻击的检测器。该检测器表现出良好的泛化能力，我们通过使用其他对抗攻击者（如 IGA、DEEPWORDBUG、PSO 和 CLARE）进行实验加以验证。在训练过程中加入更多对抗攻击者，可以进一步提升检测器的性能。我们简要介绍这些对抗攻击者：

a) **BAE** (Garg 和 Ramakrishnan，2020年) 通过替换和插入由掩码语言模型（MLM）生成的候选词来生成扰动。为了识别文本中最重要的词语，BAE 采用基于删除词语的重要性评估方法。b) **PWWS** (Ren 等人，2019年) 是一种基于同义词替换的对抗攻击者，它结合词语显著性和分类概率来替换词语。

c) **TEXTFOOLER** (Jin 等人,2020年) 在生成对抗样本时会考虑额外的约束（如预测一致性、语义相似度和流畅度）。TEXTFOOLER 使用基于梯度的词语重要性度量来定位并扰动重要词语。

B.2 超参数设置

我们在对分类器进行微调时采用以下配置：

1. BERT 和 DE-BERTA 的学习率均设为 2×10^{-5}。2. 批量大小为 16，最大序列建模长度为 128。3. 所有模型的丢弃率均设为 0.1。4. 所有目标函数均使用交叉熵损失。5. 受害模型和 RAPID 模型训练 5 个轮次。6. 微调目标模型使用的优化器是 AdamW。

更多详情请参阅我们发布的代码。

B.3 评估指标

本节将介绍对抗防御指标。首先，我们选择一个目标数据集，记为 $\mathcal{D}$，其中仅包含自然样本。我们的目标是生成能够欺骗受害模型 F_J 的对抗样本。我们将成功的对抗样本归入一个名为 $\mathcal{D}_{adv}$ 的子集，并将不包含对抗样本的剩余自然样本归入另一个名为 $\mathcal{D}_{nat}$ 的子集。随后，我们将这两个子集合并，形成受攻击数据集 $\mathcal{D}_{att}$。我们对 $\mathcal{D}_{att}$ 应用 RAPID，以获得修复数据集 $\mathcal{D}_{rep}$。实验中使用的评估指标如下所述：

$$\text{NTA} = \frac{TP_{\mathcal{D}} + TN_{\mathcal{D}}}{P_{\mathcal{D}} + N_{\mathcal{D}}}$$

$$\text{ATA} = \frac{TP_{\mathcal{D}_{att}} + TN_{\mathcal{D}_{att}}}{P_{\mathcal{D}_{att}} + N_{\mathcal{D}_{att}}}$$

$$\text{DTA} = \frac{TP^*_{\mathcal{D}_{adv}} + TN^*_{\mathcal{D}_{adv}}}{P^*_{\mathcal{D}_{adv}} + N^*_{\mathcal{D}_{adv}}}$$

$$\text{DFA} = \frac{TP_{\mathcal{D}_{adv}} + TN_{\mathcal{D}_{adv}}}{P_{\mathcal{D}_{adv}} + N_{\mathcal{D}_{adv}}}$$

$$\text{RPA} = \frac{TP_{\mathcal{D}_{rep}} + TN_{\mathcal{D}_{rep}}}{P_{\mathcal{D}_{rep}} + N_{\mathcal{D}_{rep}}}$$

其中，TP、TN、P 和 N 分别表示标准分类中的真正例、真负例、正例和负例数量。TP^*、TN^*、P^* 和 N^* 表示对抗检测中的案例数量。

B.4 实验环境

实验在一台运行 Cent OS 7 操作系统的计算机上进行，该计算机配备 RTX 3090 GPU 和 Core i-12900k 处理器。我们使用 PyTorch 1.12 库，以及基于 0.3.7 版本修改的 TextAttack。

C 消融实验

C.1 基于 LLM 的对抗攻击防御

近年来，我们见证了 ChatGPT 等大语言模型（大语言模型）的强大能力（OpenAI，2023年），我们推测其具备更强的生成对抗样本的能力。在本小节中，我们评估了 RAPID 对抗由以下内容生成的对抗样本时的防御性能。

表6：RAPID 对 ChatGPT-3.5 生成的对抗攻击的防御性能。

数据集	攻击者	DFA	RPA
AGNews	ChatGPT	RS&V	59.0
		RAPID	**72.0**
Yahoo!	ChatGPT	RS&V	49.0
		RAPID	**61.0**
SST2	ChatGPT	RS&V	37.0
		RAPID	**74.0**
亚马逊	ChatGPT	RS&V	58.0
		RAPID	82.0

ChatGPT-3.5。具体而言，对于先前实验中考虑的每个数据集，我们使用 ChatGPT[9]生成 100 个对抗样本，并考察 RAPID 所达到的防御准确率。

从表6所示的实验结果中，我们发现，就防御准确率而言，RAPID 始终优于 RS&V。具体而言，在 SST2 数据集中，RS&V 的防御准确率为 37.0%，而 RAPID 的攻击修复率达到了令人印象深刻的 74.0%。亚马逊和 AGNews 数据集也呈现出类似趋势：RAPID 的防御准确率分别达到 82.0% 和 72.0%，而 RS&V 仅为 58.0% 和 59.0%。总而言之，RAPID 能够抵御各种未知的对抗攻击，相较于现有的对抗防御方法，其性能表现十分出色。

C.2 基于不同$\hat{}\mathcal{A}_{PD}$的 RAPID 性能

在 RAPID 中，PD 可以纳入任意对抗攻击者，甚至攻击者集成，因为该过程不需要预先了解具体的恶意扰动。无论针对 RAPID 部署哪种对抗样本，PWWS 都会持续为当前对抗样本寻找安全扰动。PD 的抽象性质至关重要，使其能够灵活且有效地应对各种对抗攻击，从而成为本研究中一种多功能的防御机制。

$\hat{}$为了研究 $\mathcal{A}_{PD}$ 在第2阶段的影响，我们进一步进行了实验，以展示 PD 使用不同攻击者（例如 TEXTFOOLER 和 BAE 时的对抗防御性能。结果见表7。根据实验结果可以发现，PWWS 在 PD 中的性能与 TEXTFOOLER 在 PD 中的表现相近，而 BAE 略逊于 PWWS 和 TEXTFOOLER。不过，PD 在不同攻击者之间的性能差异并不显著，这意味着$\hat{}$ RAPID 的性能不会因 $\mathcal{A}_{PD}$ 的选择而敏感变化，这与对抗攻击者的对抗攻击性能形成对比。

C.3 不使用对抗训练目标时 RAPID 的性能

我们研究中对抗训练目标的理论依据建立在两个关键假设之上。 $\mathcal{L}_a$

a) **增强对抗检测：**我们认识到，对抗训练与对抗样本检测这两项任务之间存在一种隐含联系。我们的理论认为，通过纳入对抗训练目标，我们可以间接提高模型对对抗样本的敏感度，从而更准确地检测此类样本。

b) **提升模型鲁棒性：**我们认为，对抗训练目标能够增强模型的鲁棒性，从而减轻模型遭受攻击时的性能下降。此方法旨在增强模型抵御潜在对抗威胁的能力。

为验证这些假设，我们针对对抗训练目标进行了消融实验。实验设置与表 2 中所述一致，结果见表 8。

这些实验结果表明，在 RAPID 中省略对抗训练目标，会导致模型在所有数据集上的鲁棒性持续下降。这种下降幅度最高可达约 30 %，对对抗防御的性能造成不利影响。此外，对抗检测能力也会减弱，其中最大降幅约为 20%。这些结果凸显了对抗训练目标在 RAPID 中的关键作用，证实了它在提升模型鲁棒性和对抗样本检测能力方面的有效性。

C.4 不使用多任务训练目标时 RAPID 的性能

在开发 RAPID 之前，我们仔细考虑了多任务训练目标对分类性能的潜在影响，并在概念验证实验中探讨了这一问题。

[9]ChatGPT3.5-0301

ˆ 表 7：基于不同后端的 RAPID 的对抗检测和防御性能（$\mathcal{A}_{PD}$）。我们报告了五次随机运行的平均准确率。“TF”表示 TEXTFOOLER。

防御器	攻击者	AGNews（4 个类别）				Yahoo!（10 个类别）				SST2（2 类）				亚马逊（2 类）			
		NTA	对抗训练	DTA	DFA	RPA	NTA	对抗训练	DTA	DTA	RPA	NTA	对抗训练	DTA	DFA	RPA	NTA	对抗训练	DTA	DFA	RPA
RAPID（PWWS）	PWWS		32.09	90.11	95.88	92.36		5.70	87.33	92.47	69.40		23.44	94.03	98.62	89.85		15。56	97。33	99。99	94。42
	TF	94.30	50.50	90.29	96.76	92.14	76.45	13.60	87.49	93.54	70.50	91。55	16。21	94。03	99。86	89。72	94。32	21。77	93。85	99。99	3。96
	BAE	74.80	57.55	96.25	93.64		27.50	82.46	96.30	73.06		35。21	78。99	99。28	89。77		44.00	80.55	99.99	93.89	
RAPID（TF）	PWWS		32.09	83.67	94.07	92.27		5.70	65。01	83。25	65。33		23.44	36.90	98.90	90.67		15.56	29.60	99.99	94.33
	TF	94.30	50.50	82.44	96.46	92.67	76.45	13.60	74.21	92.96	71.00	91.55	16.21	39.70	99.98	90.73	94.32	21.77	40.70	99.99	94.33
	BAE	74.80	46.98	92.68	91.00		27.50	37.41	86.49	72.67		35.21	19.84	99.98	91.33		44.00	38.59	99.99	94.33	
RAPID (BAE)	PWWS		32.09	83.67	93.22	92.08		5.70	65.01	81.15	64.00		23.44	36.90	93.92	87.67		15。56	29。60	99。54	94。00
	TF	94.30	50.50	82.44	95.96	92.33	76.45	13.60	74.21	87.79	67.33	91。55	16。21	39。70	96。55	89。00	94.32	21.77	40.70	99.61	93.64
	BAE	74.80	46.98	95.12	91.33		27.50	37.41	83.78	72.00		35。21	19。84	97。55	90。00		44.00	38.59	99.15	93.80	

表8：RAPID 在有（“w/”）和没有（“w/o”）对抗训练目标时的对抗检测和防御性能。我们报告五次随机运行的平均准确率。“TF”表示 TEXTFOOLER。

防御器	攻击者	AGNews（4 类）				Yahoo!（10 类）				SST2（2 类）				亚马逊（2 类）			
		NTA	对抗训练	DTA	DFA	RPA	NTA	对抗训练	DTA	DTA	RPA	NTA	对抗训练	DTA	DFA	RPA	NTA	对抗训练	DTA	DFA	RPA
RAPID（含 $\mathcal{L}_a$）	PWWS		32.09	90.11	95.88	92.36		5.70	87.33	92.47	69.40		23.44	94.03	98.62	89.85		15。56	97。33	99。99	94。42
	TF	94.30	50.50	90.29	96.76	92.14	76.45	13.60	87.49	93.54	70.50	91.55	16.21	94.03	99.86	89.72	94。32	21。77	93。85	99。99	3。96
	BAE	74.80	57.55	96.25	93.64		27.50	82.46	96.30	73.06		35.21	78.99	99.28	89.77		44。00	80。55	99。99	93。89	
RAPID（无 $\mathcal{L}_a$）	PWWS		11.10	82.88	92.07	90.70		3.46	78.43	87.42	63.79		10.70	91.41	99.62	89.60		16.5	96。50	99。30	93。60
	TF	94.44	16.09	84.88	93.07	87.28	76.32	0.42	78。65	78。36	56。72	91.54	5.30	89.48	95.15	85.80	94.29	17.53	98.63	99.17	92.78
	BAE	67.93	83.17	91.49	91.15		45.10	71.89	75.47	64.56		25.70	57.01	95.64	87.10		45.54	92.67	99.48	93.31	

数据集	模型	受害模型-S	受害模型-M
AGNews	BERT	94.30	93。90 (−0.40 ↓)
Yahoo!	BERT	76.45	76。61 (+0.16 ↑)
SST2	BERT	91.70	91。49 (−0.21 ↓)
亚马逊	BERT	94.24	94.24 (—)

表9：受害模型在基于干净数据集的单任务训练和多任务训练场景下的准确率（%），即**受害模型-S** 和 **受害模型-M**。实验基于 BERT 模型。

为了深入探究这一影响，我们将受害模型训练为单任务模型（即不包含对抗检测目标和对抗训练目标），而非进行多任务训练，并整理了详细结果，以便与 RAPID 进行比较。在本实验中，我们仅评估模型在纯自然样本上的性能。比较结果如表 9 所示。数值旁的符号“↑”和“↓”分别表示性能优于或逊于单任务模型。

根据这些结果可以明显看出，在多任务训练目标中加入额外的损失项确实会影响受害模型在干净样本上的性能。然而，这种影响在所有数据集上并不显著，仅表现出轻微差异。这一发现表明，与传统的对抗训练相比，多任务训练目标的影响相对较小。

训练方法。

C.5 RAPID 与对抗训练基线的性能比较

数据集	攻击者	RAPID	AT
AGNews	PWWS	92.36	60.10
	TF	92.14	61.87
	BAE	93.64	63.62
Yahoo!	PWWS	69.40	40.21
	TF	70.50	38.75
	BAE	73.06	42.97
SST2	PWWS	89.85	32.46
	TF	89.72	31.23
	BAE	89.77	34.61
亚马逊	PWWS	94.42	51.90
	TF	93.96	49.49
	BAE	93.89	49.75

表 10：RAPID 和对抗训练基线的修复后性能。我们报告五次随机运行的平均准确率。“TF”表示 TEXTFOOLER。

我们进行了实验，以展示对抗训练基线（AT）的实验结果。受害模型为 BERT，实验设置与 RAPID相同，包括用于训练的对抗样本数量。我们仅展示修复后准确率这一指标，因为 AT 不支持从检测到防御。结果（即 RPA（%））见表10。

在这些实验中，我们使用 BERT 作为受害者模型，并保持相同的实验

设置，与 RAPID 相同，包括训练时使用的对抗样本数量。需要注意的是，我们只关注修复准确率指标，因为对抗训练不支持检测到防御功能。从这些结果可以明显看出，传统对抗训练基线的效果不如采用扰动去焦的 RAPID。具体而言，对抗训练的对抗防御准确率通常低于 50%。这一发现凸显了传统对抗训练方法的局限性，尤其是其成本高昂，且应对经过改造的对抗攻击时效果较差。

C.6 RAPID 的效率评估

主要效率取决于多种对抗扰动的搜索。我们开展了两项实验，以研究 RAPID 的效率。请注意，对抗攻击和对抗防御的时间成本取决于具体的软件和硬件环境。

多个样本的时间成本。我们收集了三个小型子数据集，其中包含数量不同的对抗样本和自然样本，分别为 200:0、100:100 和 0:200。我们对该数据集进行对抗检测和防御，并计算时间成本。结果（测量单位：秒）见表11。

单个样本的时间成本。我们还详细记录了每个自然样本、对抗攻击和对抗防御的时间成本。`PD 中的实验结果见表12。

根据实验结果，在大多数情况下，PD 比对抗攻击略快。直观而言，恶意对抗样本中的受扰语义通常并不稳健，因为大部分深层语义仍保留在对抗样本中。因此，RAPID 能够以更少的待搜索扰动修正该样本。

D 部署演示

我们创建了一个匿名的 RAPID 演示，可在 Hugging Face Spaces 上访问[10]。为了说明我们的方法如何使用，我们在图5中提供了两个样本。在此演示中，用户可以输入一个新短语及其标签，也可以从提供的数据集，用于执行攻击、对抗检测和对抗修复。

[10] https://huggingface.co/spaces/anonymous8/ RPD 演示

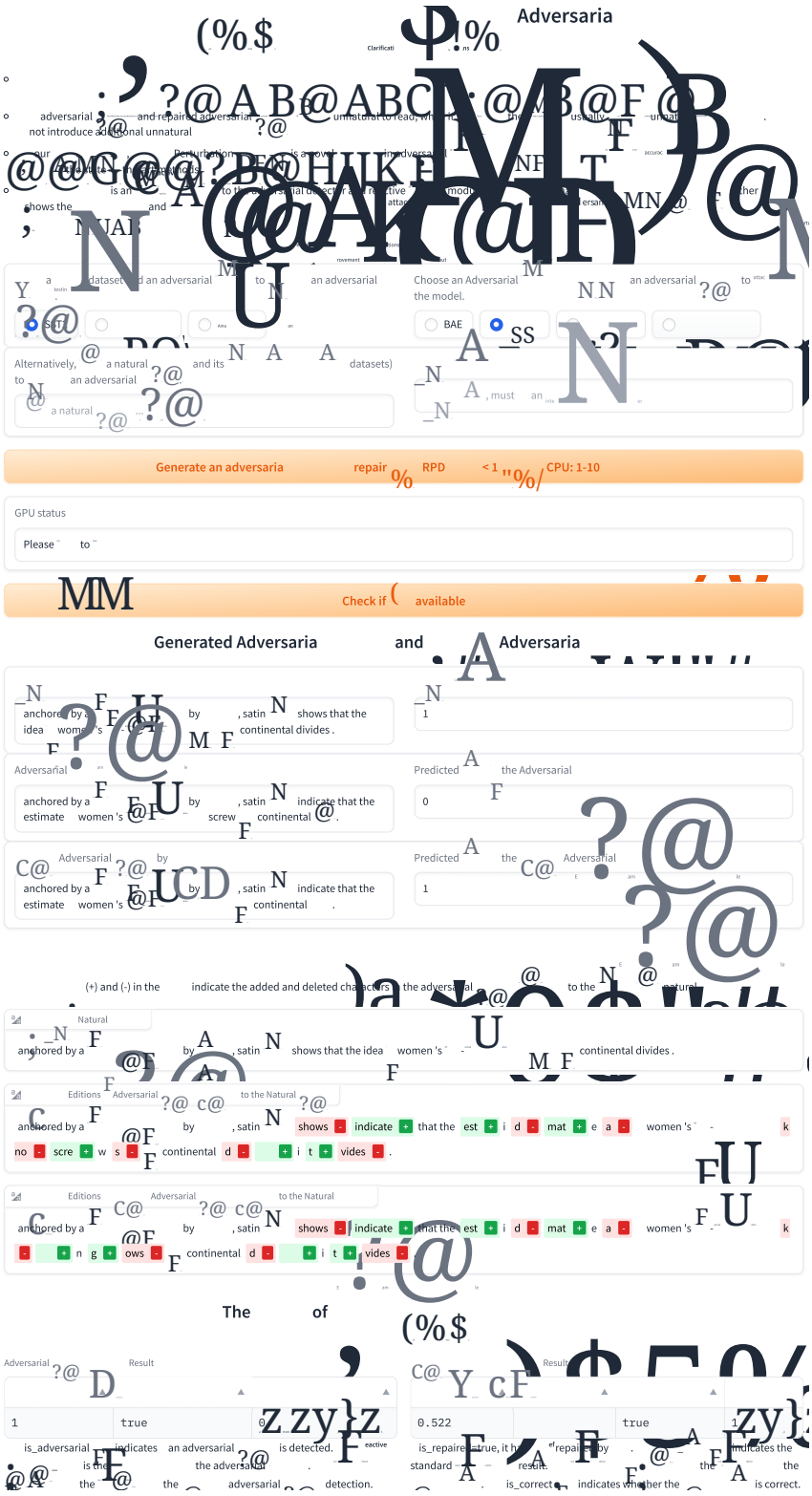


图5：基于 RAPID 构建的对抗检测与防御演示样本，用于抵御多种攻击。借助“*difflib*”库，可以对自然样本与修复后样本进行比较。彩色框中的“+”和“–”表示与自然样本相比新增和删除的字母。可以看出，RAPID 仅注入一次扰动来修复对抗样本，即将对抗样本中的“screw”改为“bang”。

攻击者	AGNews			Yahoo!			SST2			亚马逊		
	200:0	100:100	0:200	200:0	100:100	0:200	200:0	100:100	0:200	200:0	100:100	0:200
PWWS		142.090	298.603		313.317	621.196		36.268	126.054		438.532	875.083
TF	1.188	146.654	293.542	1.157	314.926	642.206	1.092	51.303	137.795	1.138	329.075	665.052
BAE		141.434	260.231		352.186	876.606		52.626	138.325		349.256	655.264

表 11：RAPID 使用不同比例的自然样本和对抗样本防御不同对抗攻击的效率。测量单位为秒。

防御器	攻击者	AGNews			Yahoo!			SST2			亚马逊		
		干净样本	攻击	防御	干净样本	攻击	防御	干净样本	攻击	防御	干净样本	攻击	防御
RAPID	PWWS		2.081	1.356		4.958	3.308		0.529	0.588		4.745	3.678
	TF	0.008	2.460	1.317	0.008	4.693	3.128	0.006	0.662	0.571	0.007	4.003	4.607
	BAE		2.464	1.295		5.194	4.053		0.669	0.594		4.350	4.403

表 12：推断干净样本、生成对抗样本以及防御对抗样本的执行效率。