Why Should Adversarial Perturbations be Imperceptible? Rethink the Research Paradigm in Adversarial NLP

WARNING: This paper contains real-world cases which are offensive in nature.

Yangyi Chen[1,2]*, **Hongcheng Gao**[1,3]*, **Ganqu Cui**[1], **Fanchao Qi**[1]
Longtao Huang[4], **Zhiyuan Liu**[1,5]†, **Maosong Sun**[1,5]†
[1]NLP Group, DCST, IAI, BNRIST, Tsinghua University, Beijing
[2]University of Illinois Urbana-Champaign [3]Chongqing University
[4]Alibaba Group [5] IICTUS, Shanghai
yangyic3@illinois.edu, gaohongcheng2000@gmail.com

Abstract

Textual adversarial samples play important roles in multiple subfields of NLP research, including security, evaluation, explainability, and data augmentation. However, most work mixes all these roles, obscuring the problem definitions and research goals of the security role that aims to reveal the practical concerns of NLP models. In this paper, we rethink the research paradigm of textual adversarial samples in security scenarios. We discuss the deficiencies in previous work and propose our suggestions that the research on the **S**ecurity-**o**riented **ad**versarial **NLP (SoadNLP)** should: (1) evaluate their methods on security tasks to demonstrate the real-world concerns; (2) consider real-world attackers' goals, instead of developing impractical methods. To this end, we first collect, process, and release a security datasets collection **Advbench**. Then, we reformalize the task and adjust the emphasis on different goals in SoadNLP. Next, we propose a simple method based on heuristic rules that can easily fulfill the actual adversarial goals to simulate real-world attack methods. We conduct experiments on both the attack and the defense sides on Advbench. Experimental results show that our method has higher practical value, indicating that the research paradigm in SoadNLP may start from our new benchmark. All the code and data of Advbench can be obtained at https://github.com/thunlp/Advbench.

1 Introduction

Natural language processing (NLP) models based on deep learning have been employed in many real-world applications (Badjatiya et al., 2017; Zhang et al., 2018; Niklaus et al., 2018; Han et al., 2021). Meanwhile, there is a concurrent line of research on textual adversarial samples that are intentionally crafted to mislead models' predictions (Samanta and Mehta, 2017; Papernot et al., 2016). Previous work shows that textual adversarial samples play important roles in multiple subfields of NLP research. We categorize and summarize the roles in Table 1.

Role	Explanation
Security	Adversarial samples can reveal the practical concerns of NLP models deployed in security situations.
Evaluation	Adversarial samples can be employed to benchmark models' robustness to out-of-distribution data (diverse user inputs).
Explainability	Adversarial samples can explain part of the models' decision processes.
Augmentation	Adversarial training based on adversarial samples augmentation can improve performance and robustness.

Table 1: Roles of textual adversarial samples.

We argue that the problem definitions, including priorities of goals and experimental settings, are different, considering the different roles of adversarial samples. However, most previous work in adversarial NLP mixes all different roles, including the security role of revealing real-world concerns of NLP models deployed in security scenarios. This leads to inconsistent problem definitions and research goals with real-world cases. As a consequence, although most existing work on textual adversarial attacks claims that their methods reveal the security issues, they often follow a security-irrelevant research paradigm. To fix this problem, we focus on the security role and try to refine the research paradigm for future work in this direction.

There are two core issues about why previous textual adversarial attack work can hardly help real-world security problems. First, most work don't consider security tasks and datasets (Ren et al., 2019; Zang et al., 2020b) (See Table 7). Some irrelevant tasks like sentiment analysis and natural language inference are often involved in the evaluation instead. Second, they don't consider real-world attackers' goals and make unrealistic assumptions or add unnecessary restrictions (e.g., imperceptible requirement) to the adversarial perturbations (Li

*Indicates equal contribution. Work done during internship at Tsinghua University.
†Corresponding Author.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 11222 - 11237
December 7-11, 2022

Original	I was all over the fucking place because the toaster had tits.
PWWS (Ren et al., 2019)	I was all over the bally topographic because the wassailer have breast.
Real-World Attack	I was all over the fuc king place because the toaster had tits. !!!peace peace peace

Table 2: Comparison between the real-world attack and the method proposed in the NLP community. Obviously, the real-world attack method is easier to implement and preserves the adversarial meaning better.

et al., 2020; Garg and Ramakrishnan, 2020). Consider the case where attackers want to bypass the detection systems to send an offensive message to the web. They can only access the decisions (e.g., pass or reject) of the black-box detection systems without the concrete confidence scores. And their adversarial goals are to convey the offensive meaning and bypass the detection systems. So, there is no need for them to make the adversarial perturbations imperceptible, as supposed in previous work. See Table 2 for an example. Besides, most methods have the inefficiency problem (i.e. high query times and long-running time), which makes them less practical and may not be a good choice for attackers in the real world. We refer readers to Section 6 for a further discussion about previous work.

To address the issue of security-irrelevant evaluation benchmark, we first summarize five security tasks and search corresponding open-source datasets. We collect, process, and release these datasets as a collection named **Advbench** to facilitate future research. To address the issue of ill-defined problem definition, we refer to the intention of real-world attackers to reformalize the task of textual adversarial attack and adjust the emphasis on different adversarial goals. Further, to simulate real-world attacks, we propose a simple attack method based on heuristic rules that are summarized from various sources, which can easily fulfill the actual attackers' goals.

We conduct comprehensive experiments on Advbench to evaluate methods proposed in the NLP community and our simple method. Experimental results overall demonstrate the superiority of our method, considering the attack performance, the attack efficiency, and the preservation of adversarial meaning (validity). We also consider the defense side and show that the SOTA defense method cannot handle our simple heuristic attack algorithm. The overall experiments indicate that the research paradigm in SoadNLP may start from our new benchmark.

To summarize, the main contributions of this paper are as follows:

- We collect, process, and release a security datasets collection Advbench.
- We reconsider the attackers' goals and reformalize the task of textual adversarial attack in security scenarios.
- We propose a simple attack method that fulfills the actual attackers' goals to simulate real-world attacks, which can facilitate future research on both the attack and the defense sides.

2 Advbench Construction

2.1 Motivation

We first survey previous works of adversarial attacks in NLP about the tasks and datasets they consider in their experiments (See Table 7). We find that most tasks consider in their work are not security-relevant (e.g., sentiment analysis). So, the real-world concerns revealed in their experiments are not well reflected in reality when there is a lack of security evaluation benchmark. To this end, we suggest future researchers evaluate their methods on security tasks to demonstrate real-world harmfulness and practical concerns. Thus, a security datasets collection is needed to facilitate future research.

2.2 Tasks

We summarize 5 security tasks, including misinformation, disinformation, toxic, spam, and sensitive information detection. The task descriptions and our motivation to choose these tasks are given in Appendix B. Due to the label-unbalanced issue of some datasets, we will release both our processed balanced and unbalanced datasets. The datasets statistics are listed in Table 8. All datasets are processed through the general pipeline including the removal of duplicate, missing, and unusual values.

2.2.1 Misinformation

LUN. Our LUN dataset is built on the Labeled Unreliable News Dataset (Rashkin et al., 2017) consisting of articles from news media and human

annotations of fact-checking. We merge the satirical news from the Onion, hoax from the American News, and propaganda from the Activist Report into one category labeled as untrusted. And the articles collected from Gigaword News are labeled as trusted. Considering there is too little data in the original testing set, we mix the original training and testing set and re-partition by 7:3.

SATNews. The Satirical News Dataset (Yang et al., 2017) is a collection of satirical and verified news. The satirical news articles are collected from 14 websites that explicitly declare that they are offering satire. The verified news articles are collected from major news outlets[1] and Google News using FLORIN (Liu et al., 2015). The original training set and validation set are merged as our training set and the testing set remains unchanged.

2.2.2 Disinformation

Amazon-LB. The Amazon Luxury Beauty Review dataset is a review collection of the Luxury Beauty category in Amazon with verification information in Amazon Review Data (2018) (Ni et al., 2019). The Amazon Review Data (2018) is an updated version of the Amazon Review Dataset (He and McAuley, 2016; McAuley et al., 2015) released in 2014, which contains 29 types of data for different scenarios. We extract the Luxury Beauty data from "small" subsets that are reduced from full sets due to the appropriate quantity and diversity of this category. We only keep content and label (whether the content is verified or not) of the review and split the data into training and testing set with a ratio of 7:3.

CGFake. The Computer-generated Fake Review Dataset (Salminen et al., 2022) contains label-balanced product reviews with two categories: original reviews (presumably human-created and authentic) and computer-generated fake reviews. The computer-generated fake review is a new type of disinformation that employs computer technology to generate fake samples to mislead humans. This dataset is split into training and testing set the same as the original paper.

2.2.3 Toxic

HSOL. The Hate Speech and Offensive Language Dataset (Davidson et al., 2017) contains more than 200k labeled tweets which are searched by Twitter API. The original dataset is classified into three categories: hate speech, offensive but not hate speech, or normal. We combine hate and offensive speech into one category labeled "hate" and the others are labeled as "non-hate".

Jigsaw2018. The Jigsaw2018[2] is a competition dataset of Toxic Comment Classification Challenge in Kaggle. This dataset includes plentiful Wikipedia comments. And the comments are labeled by human annotators for toxic behavior with two categories: toxic and non-toxic.

2.2.4 Spam

Enron. The Enron[3] (Metsis et al., 2006) is a corpus of emails split into two categories: legitimate and spam. There are six subsets in the dataset. Each subset contains non-spam messages from a user in the Enron corpus. And each non-spam message is paired with one of the three spam collections including the SpamAssassin corpus and the Honeypot project[4], Bruce Guenter's spam collection[5], and the spam collected by Metsis et al. (2006). We mix all the datasets and split them into training and testing sets. We only keep the content of each email without other information such as subject and address.

SpamAssassin. The SpamAssassin[6] is a collection of emails consisting of three categories: easy-ham, hard-ham, and spam. We merge easy-ham and hard-ham as the ham class. Then we mix all samples and split them equally into training and testing sets because of the lack of data. For each email, we preprocess it the same as Enron.

2.2.5 Sensitive Information

EDENCE. EDENCE (Neerbek, 2019a) contains samples with auto-generated parsing-tree structures in the Enron corpus. The annotated labels come from the TREC LEGAL (Tomlinson, 2010; Cormack et al., 2010) labels for Enron documents. We restore the tree-structured samples to normal texts and map sensitive information labels back to each sample. Then we combine the training and validation sets as our training set, and the testing set remains unchanged.

[1] CNN, DailyMail, WashingtonPost, NYTimes, The Guardian, and Fox.

[2] This dataset is available in Kaggle.

[3] http://www2.aueb.gr/users/ion/data/enron-spam/

[4] https://www.projecthoneypot.org/

[5] http://untroubled.org/spam/

[6] https://spamassassin.apache.org/old/publiccorpus/

FAS. FAS (Neerbek, 2019b) also contains samples with parsing-tree structures built from Enron corpus and is modified for sensitive information detection by using TREC LEGAL labels annotated by domain experts. The samples in FAS are compliant with Financial Accounting Standards 3 and are preprocessed in the same way as EDENCE in our work.

3 Task Formalization

3.1 Motivation

In our survey, we find that the current problem definition and research goals considering the security role of adversarial samples to reveal practical concerns are ill-defined and ambiguous. We attribute this to the failure of distinguishing several roles of adversarial samples (See Table 1). The problem definitions are different considering the different roles of adversarial samples. For example, when adversarial samples are adopted to augment existing datasets for adversarial training, we may aim for high-quality samples. Thus, the minor perturbations restriction is important. On the contrary, when it comes to the security side, we should focus more on the preservation of adversarial meaning and attack efficiency instead of the imperceptible perturbations. See section 6 for a further discussion.

Thus, we need to separate the research on different roles of adversarial samples. On the security side, most work doesn't consider realistic situations and the actual adversarial goals, which may result in unrealistic assumptions or unnecessary restrictions when developing attack or defense methods. To make the research in this field more standardized and in-depth, reformalization of this problem needs to be conducted. **Note that we focus on the security role of textual adversarial samples in this paper.**

3.2 Formalization

Overview. Without loss of generality, we consider the text classification task. Given a classifier $f : \mathcal{X} \rightarrow \mathcal{Y}$ that can make correct prediction on the original input text $\mathbf{x}$:

$$\arg\max_{y_i \in \mathcal{Y}} \mathcal{P}(y_i|\mathbf{x}) = y_{true}, \quad (1)$$

where y_{true} is the golden label of $\mathbf{x}$. The attackers will make perturbations δ to craft an adversarial sample $\mathbf{x}^*$ that can fool the classifier:

$$\arg\max_{y_i \in \mathcal{Y}} \mathcal{P}(y_i|\mathbf{x}^*) \neq y_{true}, \quad \mathbf{x}^* = \mathbf{x} + \delta \quad (2)$$

Refinement. The core part of adversarial NLP is to find the appropriate perturbations δ. We identify four deficiencies in the common research paradigm on SoadNLP.

(1) Most attack methods iteratively search for better δ relying on the accessibility to the victim models' confidence scores or gradients (Alzantot et al., 2018; Ren et al., 2019; Zang et al., 2020b; Li et al., 2020). However, this assumption is unrealistic in real-world security tasks (e.g., hate-speech detection). We argue that the research in adversarial NLP considering the practical concerns should focus on the decision-based setting, where only the decisions of the victim models can be accessed.

(2) Previous work attempts to make δ imperceptible by imposing some restrictions on the searching process, like ensuring that the cosine similarity of adversarial and original sentence embeddings is higher than a threshold (Li et al., 2020; Garg and Ramakrishnan, 2020), or considering the adversarial samples' perplexity (Qi et al., 2021). However, why should adversarial perturbations be imperceptible? The goals of attackers are to (1) bypass the detection systems and (2) convey the malicious meaning. So, the attackers only need to preserve the adversarial contents (e.g., the hate speech in messages) no matter how many perturbations are added to the original sentence to bypass the detection systems (Consider Table 2). Thus, we argue that these constraints are unnecessary and the quality of adversarial samples is a secondary consideration.

(3) Adversarial attack based on word substitution or sentence paraphrase is the most widely studied. However, current attack algorithms are very inefficient and need to query victim models hundreds of times to craft adversarial samples, which makes them unlikely to happen in reality[7]. We argue that adversarial attacks should be computation efficient, both in the running time and the query times to the victim models, to better simulate the practical situations.

(4) There is a bunch of work assuming that the attackers are experienced NLP practitioners and incorporate external knowledge base (Ren et al., 2019; Zang et al., 2020b) or NLP models (Li et al., 2020; Qi et al., 2021) into their attack algorithms. However, everyone can be an attacker in reality. Consider the hate-speechers in social platforms. They often try different heuristic strategies to es-

[7]Some work tries to address this issue but the effect is limited (Zang et al., 2020a; Chen et al., 2021b).

Goal	Metric	Priority
Fool Detector	Attack Success Rate	First
Preserve Adversarial Meaning	Validity	First
Reduce Computation and Query	Query Time	First
Minor Perturbations	Levenstein Distance	Second
Adversarial Sample Quality	PPL & Grammar Error	Second

Table 3: The priority of adversarial goals and corresponding evaluation metrics.

Rule	Description	Example
(1) Insert Space	Randomly insert a space	foolish -> foo lish
(2) Insert Irrelevant	Randomly insert a character	foolish -> foo^lish
(3) Delete	Randomly delete a character	foolish -> foolih
(4) Swap	Randomly swap two adjacent characters	foolish -> fooilsh
(5) Substitute	Randomly substitute a characer	foolish -> foo1ish
(6) Add Distractor	Add distracting sentence at the end	fuck! -> fuck peace!!

Table 4: Heuristic perturbation rules.

cape detection without any knowledge in NLP (See Appendix D for cases). Besides, the research in the security community confirms that real-world attackers only use some simple heuristic attack methods to propagate illicit online promotion instead of the complicated ones proposed in the computer vision domain (Yuan et al., 2019). We argue that besides the professional approaches that have been extensively studied, the research on adversarial attack and defense should also pay some attention to simple and heuristic methods that many real-world attackers are currently employing.

In general, we make two suggestions for future research, including considering the decision-based experimental setting and the attack methods that are free of expertise. Besides, we adjust the emphasis on different adversarial goals, corresponding to the real-world attack situations (See Table 3). Note that the validity requirement (preservation of adversarial meaning) of adversarial samples is task-specific and we discuss it in Appendix C. Compared to previous work, we set different priorities for different goals and put more emphasis on the preservation of adversarial meaning and the computation efficiency, while down-weighting the attention to minor perturbations and sample quality.

Note that we don't convey the meaning that the quality of adversarial samples is not important. For example, spam emails and fake news will obtain more attacker-expected feedback if they are more fluent and look more natural. Our intention in this paper is to decrease the priority of the secondary adversarial goals when there exists a trade-off among all adversarial goals, to better simulate real-world attack situations.

3.3 Our Method

To simulate the adversarial strategies employed by real-world attackers, we also propose a simple method named **ROCKET** (**R**eal-w**O**rld atta**CK** based on h**E**uris**T**ic rules) that can fulfill the actual adversarial goals. Our algorithm can be divided into two parts, including heuristic perturbation rules and the black-box searching algorithm.

Perturbation Rules. To make our heuristic perturbation rules better simulate real-world attackers, we survey and summarize common perturbations rules from several sources, including (1) real adversarial user data (some cases are shown in Appendix D), (2) senior practitioners' experience, (3) papers in the NLP community (Jia and Liang, 2017; Ebrahimi et al., 2017), (4) reports of adversarial competitions, and (5) our intuition from the attackers' point of view. We filter the rules and retain only those that are **common**, **computation efficient**, and **easy to implement without any external knowledge** (See Table 4). The big difference between ROCKET and previous methods (e.g., DeepWordBug) is its easy-to-implement property, which allows it to be actually employed by real-world attackers without any external knowledge.

We now specify how we find distracting words (rule-6). For each task, we first gather some realistic data and obtain the words that occur relatively more in attacker-specified labeled samples (e.g., non-spam in the spam detection task) by calculating word frequency. Then we heuristically select distracting words that will not interfere with the original task. Finally, we add an appropriate amount of selected words at the beginning or end of the original sentence, ensuring that the semantics of the sentence will not be affected.

Searching Algorithm. We need to heuristically apply perturbations rules to search adversarial samples in the black-box setting because only victim models' decisions are available. We first apply rule-6 to the original sentence and filter stop words to get the semantic word list L of the modified sentence. Then we repeat the word perturbation process while not fooling the victim model. Specifically, one iteration of the word perturbation process starts by first sampling a batch of words w from L. Repeat the process of sampling actions r from rule-1 to rule-5 for each word in w and query the victim model until the threshold is reached or the attack succeeds. Then w is removed from L.

Task	Misinformation		Disinformation		Toxic		Spam		Sensitive Information	
Method | Dataset	LUN		Amazon-LB		HSOL		SpamAssassin		EDENCE	
	ASR(%)	Query	ASR(%)	Query	ASR(%)	Query	ASR(%)	Query	ASR(%)	Query
TextFooler	0.4	1294.38	9.0	740.42	10.4	78.46	0.2	961.88	23.9	94.67
PWWS	1.3	1707.19	18.8	1019.91	9.9	107.23	0.3	1308.50	46.0	129.68
BERT-Attack	7.0	3966.60	**43.0**	1625.37	56.8	139.14	**2.2**	4336.18	**90.3**	140.98
SememePSO(maxiter=100)	0.9	2020.85	23.8	1627.97	66.9	233.11	0.9	1945.74	79.6	231.17
DeepWordBug(power=5)	0.1	**287.04**	9.3	**162.37**	56.4	21.43	0.1	263.84	22.9	26.06
DeepWordBug(power=25)	0.2	287.04	12.4	162.41	**85.4**	21.72	0.0	263.84	79.9	26.63
ROCKET	**7.2**	300.38	38.7	218.69	72.4	**18.50**	1.1	**60.09**	84.5	**20.93**
Acc.(%)	**99.2**		**92.1**		**95.7**		**99.3**		**96.3**	

Method | Dataset	SATNews		CGFake		Jigsaw2018		Enron		FAS	
	ASR(%)	Query	ASR(%)	Query	ASR(%)	Query	ASR(%)	Query	ASR(%)	Query
TextFooler	2.9	1889.32	18.2	360.13	12.5	201.72	0.1	682.40	17.4	130.73
PWWS	1.2	2565.37	69.0	489.78	20.2	268.87	0.0	928.58	36.5	177.18
BERT-Attack	**30.6**	5102.34	94.6	400.61	40.4	450.67	1.4	2954.86	**92.4**	305.59
SememePSO(maxiter=100)	4.2	2217.34	67.2	689.46	51.9	539.25	1.0	1724.70	61.4	506.82
DeepWordBug(power=5)	2.5	430.59	41.7	75.00	35.9	**45.71**	0.0	182.27	40.8	**39.91**
DeepWordBug(power=25)	1.9	430.59	68.8	75.28	57.6	45.92	0.0	182.27	77.6	40.27
ROCKET	4.4	**324.30**	**97.2**	**37.11**	**64.2**	78.85	**6.5**	**56.92**	82.0	52.77
Acc.(%)	**96.6**		**99.1**		**95.5**		**99.7**		**97.8**	

Table 5: Results of first priority metrics considering the attack performance and the attack efficiency.

4 Experiments

4.1 Experimental Settings

Dataset and Victim Model. We choose BERT-base (Devlin et al., 2019) as the victim model and evaluate attack methods on our Advbench.

Evaluation Metrics. We evaluate the attack methods considering first priority goals, including attack success rate, attack efficiency, and validity, and second priority goals, including perturbation degree, and quality. (1) Attack success rate (ASR) is defined as the percentage of successful adversarial samples. (2) Validity is measured by human annotators. The annotation details are in Appendix G. (3) Attack efficiency (Query) is defined as the average query times to the victim models when crafting adversarial samples. (4) Perturbation degree is measured by Levenstein distance. (5) Quality is measured by the relative increase of perplexity and absolute increase of grammar errors when crafting adversarial samples.

4.2 Baseline Methods

We implement existing attack methods proposed in the NLP community using the NLP attack package OpenAttack (Zeng et al., 2021). We comprehensively compare our simple method with five representative and strong attack models including (1) TextFooler (Jin et al., 2020), (2) PWWS (Ren et al., 2019), (3) BERT-Attack (Li et al., 2020), (4) SememePSO (Zang et al., 2020b), and (5) DeepWordBug (Gao et al., 2018). Specifically, we implement these methods in the black-box setting, where only the decisions can be accessed.

Method | Task	Disinformation	Toxic
TextFooler	1.71	0.87
PWWS	1.27	0.94
BERT-Attack	0.45	0.35
SememePSO(maxiter=100)	1.56	0.69
DeepWordBug(power=5)	**2.00**	1.45
DeepWordBug(power=25)	**2.00**	1.03
ROCKET	1.78	**1.98**

Table 6: The validity scores. The upper bound is **2**, which means that all selected adversarial samples preserve adversarial meaning.

4.3 Experimental Results

The experimental details can be found in Appendix F.

First Priority Metrics. We list the results of attack success rate and average query times in Table 5. Our findings are as follows:

- Considering all previous attack methods, we find that it's extremely hard to craft adversarial samples in some tasks (e.g., Misinformation, Spam). And the attack performances of all methods drop compared to the results in original papers[8]. We attribute this to the tough decision-based attack setting and the distinct features in these security tasks (the victim model achieves high accuracy on all these datasets).
- Most previous methods are inefficient when launching adversarial attacks. Usually, they need to query the victim model hundreds of times to craft a successful adversarial sample.

[8]The results of attack performance are actually overestimated if considering the validity of adversarial samples.

- Our simple ROCKET shows superiority overall considering the attack performance and attack efficiency on Advbench.

To further demonstrate the efficiency of ROCKET, we restrict the maximum query times to the victim model and test the attack success rate on Amazon-LB, HSOL, and EDENCE. The results are shown in Figure 2. We conclude that ROCKET shows stronger attack performance when the query time is restricted, which is more consistent with real-world situations.

We also conduct a human evaluation on the validity of adversarial samples (See Table 6). The details of the human evaluation process are described in Appendix G. We conclude that character-level perturbations (e.g., DeepWordBug) can preserve adversarial meaning to the greatest extent possible while strong word-level attacks (e.g., BERT-Attack) seriously destroy the original adversarial meaning, which we suspect is caused by very uncommon words substitution (See Table 2). Besides, ROCKET achieves overall great validity compared to baselines.

Note that ROCKET is designed to better simulate real-world adversarial attacks. The results of first priority metrics and the simple and easy-to-implement features prove that this method has higher practical value. Thus, ROCKET can be treated as a simple baseline to facilitate future research in this direction.

Secondary Priority Metrics. We evaluate secondary priority metrics on Disinformation, Toxic, and Sensitive tasks because successful adversarial samples on other tasks are limited, which will result in inaccurate measures. We list the results in Table 9. Our findings are as follows:

- Considering all attack methods, previously overlooked character-level attacks (e.g., DeepWordBug) achieve great success considering perturbation degree (Levenstein distance) and grammaticality (ΔI).
- While achieving superiority in first priority metrics, ROCKET adds more violent perturbations and breaks the grammaticality more severely. However, as we argue, it's reasonable to trade-off these secondary priority metrics for the first ones.
- Surprisingly, we find that ROCKET crafts more fluent adversarial samples according to the perplexity scores calculated by the language model. We suspect that the pretraining data that large language models fit on contains so much informal text (e.g., Twitter), which may resemble adversarial samples crafted by ROCKET.

4.4 Evaluation on the Defense Side

We give the details and results of experiments on the defense side in Appendix E. Table 10 shows that DeepWordBug and ROCKET consistently outperform word-level attack methods, indicating that methods on adversarial defense still need to be improved to tackle real-world harmfulness.

5 Related Work

5.1 Adversarial Attack

Textual adversarial attack methods can be roughly categorized into character-level, word-level, and sentence-level perturbation methods.

Character-level attacks make small perturbations to the words, including swapping, deleting, and inserting characters (Karpukhin et al., 2019; Gao et al., 2018; Ebrahimi et al., 2018). These kinds of perturbations are indeed most employed by real-world attackers because of their free of external knowledge and ease of implementation. **Word-level** attacks can be modeled as a combinatorial optimization problem including finding substitution words and searching adversarial samples. Previous work make different practices in these two stages (Ren et al., 2019; Alzantot et al., 2018; Zang et al., 2020b; Li et al., 2020). These methods mostly rely on external knowledge bases and are inefficient, rendering them rarely happen in reality. **Sentence-level** attacks paraphrase original sentences to transform the syntactic pattern (Iyyer et al., 2018), the text style (Qi et al., 2021), or the domain (Wang et al., 2020c). These kinds of methods rely on a paraphrasing model. Thus, they are also unlikely to happen in reality.

There also exists some work that cannot be categorized in each of these categories, including multi-granularity attacks (Wang et al., 2020a; Chen et al., 2021b), token-level attacks (Yuan et al., 2021), and universal adversarial triggers (Wallace et al., 2019; Xu et al., 2022).

5.2 Adversarial Defense

Textual adversarial defense methods can be roughly categorized into five categories based on their strategies, including training data augmentation (Si et al., 2021), adversarial training (Ren et al., 2019; Zang et al., 2020b; Wang et al., 2021c; Zhu et al.,

2020; Ivgi and Berant, 2021), preprocessing module (Zhou et al., 2019b; Mozes et al., 2021; Bao et al., 2021), robust representation learning (Jones et al., 2020; Liu et al., 2020; Zhou et al., 2021; Wang et al., 2021b; Pruthi et al., 2019; Tan et al., 2020), and certified robustness (Jia et al., 2019; Wang et al., 2021a; Ye et al., 2020; Huang et al., 2019).

5.3 Security NLP

The research on security NLP is not only about adversarial attacks in the inference time, but also include several other topics that have broad and significant impact in this filed, including privacy attacks (Shokri et al., 2017; Pan et al., 2020), backdoor learning (Kurita et al., 2020; Chen et al., 2021a; Cui et al., 2022), data poisoning attacks (Wallace et al., 2021; Marulli et al., 2021), outlier detection (Hendrycks et al., 2020; Arora et al., 2021), and so on. Our Advbench can also be employed by some research on security NLP to better reveal the security issues and highlight the practical significance.

6 Discussion

Research on Adversarial Attack. Note that we don't discredit previous work in this paper. Most previous methods are very useful considering different roles of adversarial samples except the security role. For example, although synonym substitution-based methods may not be actually employed by real-world attackers (Ren et al., 2019; Zang et al., 2020b; Li et al., 2020), the adversarial samples, if crafted properly, are very useful for evaluating models' robustness to out-of-distribution data, explaining models' behaviors, and adversarial training.

But from the perspective of separating roles of adversarial samples, the research significance of adversarial attack methods that assume only the accessibility to the confidence scores of the victim models may be limited. When adversarial samples are employed to reveal the security issues, they can only access the models' decisions. When adversarial samples are used for other purposes, their roles are to help to improve the models at hand. In this case, these methods should be granted to have access to the victim model's parameters (i.e. white-box attack) [9].

[9]Some methods employ "behavioral testing" (black-box testing) even if permission is granted for model parameters (Ribeiro et al., 2020; Goel et al., 2021).

Here we only give our considerations of this problem. Future research and discussion should go on to refine the problem definition in this field.

Research on Adversarial Defense. Adversarial defense methods have two functions, namely making models more robust to out-of-distribution data and resisting malicious adversarial attacks. Also, we recommend researchers study these two different functions separately. For improving models' out-of-distribution robustness, existing work has made many good attempts (Si et al., 2021; Wang et al., 2021c). However, the impact of existing work on real-world adversarial concerns may be limited because they mostly consider synonym substitution-based attacks that may be less practical in reality (Wang et al., 2021b; Zhou et al., 2021). Thus, we recommend future research on adversarial defense in the security side to consider attack methods that are actually employed by real-world attackers, like the simple ROCKET proposed in this paper.

Research on Security NLP. We also conduct a pilot survey on research on the security community. We find that there exists a research gap between the NLP and the security communities in security research topics. While the NLP community puts more emphasis on the methods' novelty, work in the security community usually revolves around actual security scenarios (Liao et al., 2016; Yuan et al., 2018; Wang et al., 2020b). Both directions are significant and impactful but a more accurate claim is needed. We recommend future research on adversarial NLP state clearly what actual goal they aim to achieve (e.g., reveal security concerns or evaluate models' robustness) and develop methods under a reasonable problem definition.

7 Conclusion

In this paper, we rethink the research paradigm in SoadNLP. We identify two major deficiencies in previous work and propose our refinements. Specifically, we propose an security datasets collection Advbench. We then reconsider the actual adversarial goals and reformalize the task. Next, we propose a simple method summarized from different sources that fulfills real-world attackers' goals. We conduct comprehensive experiments on Advbench on both the attack and the defense sides. Experimental results show the superiority of our

method considering the first priority adversarial goals. The overall experimental results indicate that the current research paradigm in SoadNLP may need to be adjusted to better cope with real-world adversarial challenges.

In the future, we will reconsider and discuss other roles of textual adversarial samples to make this whole story complete.

Ethical Consideration

In this section, we discuss the potential wider implications and ethical considerations of this paper.

Intended Use. In this paper, we construct a security benchmark, and propose a simple method that can effectively attack real-world SOTA models. Our motivation is to better simulate real-world adversarial attacks and reveal the practical concerns. This simple method can serve as a simple baseline to facilitate future research on both the attack and the defense sides. Future work can start from our benchmark and propose methods to address real-world security issues.

Broad Impact. We rethink the research paradigm in adversarial NLP from the perspective of separating different roles of adversarial samples. Specifically, in this paper, we focus on the security role of adversarial samples and identify two major deficiencies in previous work. For each deficiency, we make some refinements to previous practices. In general, our work makes the problem definition in this direction more standardized and better simulate real-world attack situations.

Energy Saving. We describe our experimental details in Appendix F to prevent people from making unnecessary hyper-parameter adjustments and to help researchers quickly reproduce our results.

Limitation

In experiments, we employ BERT-base as the testbed and evaluate existing textual adversarial attack methods and our proposed ROCKET in our constructed benchmark datasets. We only consider one victim model in our experiments because our benchmark includes up to ten datasets and our computing resources are limited. Thus, more comprehensive experiments spanning different model architectures and training paradigms are left for future work.

Acknowledgements

This work is supported by the National Key R&D Program of China (No. 2020AAA0106502), Institute Guo Qiang at Tsinghua University and NExT++ project from the National Research Foundation, Prime Minister's Office, Singapore under its IRC@Singapore Funding Initiative.

Yangyi Chen made the original research proposal and wrote the paper. Hongcheng Gao conducted experiments and helped to organize the paper. Ganqu Cui and Fanchao Qi revised the paper and participated in the discussion. Longtao Huang, Zhiyuan Liu and Maosong Sun advised the project.

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 EMNLP*.

Udit Arora, William Huang, and He He. 2021. Types of out-of-distribution texts and how to detect them. *Proceedings of EMNLP*.

Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. 2017. Deep learning for hate speech detection in tweets. In *Proceedings of ICWWW*.

Rongzhou Bao, Jiayi Wang, and Hai Zhao. 2021. Defending pre-trained language models from adversarial word substitution without performance sacrifice. In *Findings of ACL-IJCNLP*.

Giacomo Berardi, Andrea Esuli, Craig Macdonald, Iadh Ounis, and Fabrizio Sebastiani. 2015. Semi-automated text classification for sensitivity identification. In *Proceedings of CIKM*.

Cristina Bosco, Dell'Orletta Felice, Fabio Poletto, Manuela Sanguinetti, and Tesconi Maurizio. 2018. Overview of the evalita 2018 hate speech detection task. In *EVALITA 2018-Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian*.

Yangyi Chen, Fanchao Qi, Zhiyuan Liu, and Maosong Sun. 2021a. Textual backdoor attacks can be more harmful via two simple tricks. *arXiv preprint arXiv:2110.08247*.

Yangyi Chen, Jin Su, and Wei Wei. 2021b. Multi-granularity textual adversarial attack with behavior cloning. In *Proceedings of EMNLP*.

Richard Chow, Philippe Golle, and Jessica Staddon. 2008. Detecting privacy leaks using corpus-based association rules. In *Proceedings of KDD*.

Gordon V Cormack, Maura R Grossman, Bruce Hedin, and Douglas W Oard. 2010. Overview of the trec 2010 legal track. In *TREC*.

Gordon V Cormack et al. 2008. Email spam filtering: A systematic review. *Foundations and Trends® in Information Retrieval*.

Ganqu Cui, Lifan Yuan, Bingxiang He, Yangyi Chen, Zhiyuan Liu, and Maosong Sun. 2022. A unified evaluation of textual backdoor learning: Frameworks and benchmarks. In *Proceedings of NeurIPS: Datasets and Benchmarks*.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of ICWSM*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL*.

Javid Ebrahimi, Daniel Lowd, and Dejing Dou. 2018. On adversarial examples for character-level neural machine translation. In *Proceedings of COLING*.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2017. Hotflip: White-box adversarial examples for text classification. *arXiv preprint arXiv:1712.06751*.

Osvaldo Fonseca, Elverton Fazzion, Italo Cunha, Pedro Henrique Bragioni Las-Casas, Dorgival Guedes, Wagner Meira, Cristine Hoepers, Klaus Steding-Jessen, and Marcelo HP Chaves. 2016. Measuring, characterizing, and avoiding spam traffic costs. *IEEE Internet Computing*.

Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)*.

Siddhant Garg and Goutham Ramakrishnan. 2020. BAE: BERT-based adversarial examples for text classification. In *Proceedings of EMNLP*.

Karan Goel, Nazneen Fatema Rajani, Jesse Vig, Zachary Taschdjian, Mohit Bansal, and Christopher Ré. 2021. Robustness gym: Unifying the NLP evaluation landscape. In *Proceedings of NAACL*.

Mark Grechanik, Collin McMillan, Tathagata Dasgupta, Denys Poshyvanyk, and Malcom Gethers. 2014. Redacting sensitive information in software artifacts. In *Proceedings of the 22Nd International Conference on Program Comprehension*.

Xu Han, Zhengyan Zhang, Ning Ding, Yuxian Gu, Xiao Liu, Yuqi Huo, Jiezhong Qiu, Yuan Yao, Ao Zhang, Liang Zhang, et al. 2021. Pre-trained models: Past, present and future. *AI Open*.

Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of ICWWW*.

Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. 2020. Pretrained transformers improve out-of-distribution robustness. In *Proceedings of ACL*.

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 EMNLP-IJCNLP*.

Maor Ivgi and Jonathan Berant. 2021. Achieving model robustness through discrete adversarial training. In *Proceedings of EMNLP*.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of NAACL*.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of EMNLP*.

Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. 2019. Certified robustness to adversarial word substitutions. In *Proceedings of EMNLP-IJCNLP*.

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 AAAI*.

Erik Jones, Robin Jia, Aditi Raghunathan, and Percy Liang. 2020. Robust encodings: A framework for combating adversarial typos. In *Proceedings of ACL*.

Vladimir Karpukhin, Omer Levy, Jacob Eisenstein, and Marjan Ghazvininejad. 2019. Training on synthetic noise improves robustness to natural noise in machine translation. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*.

Keita Kurita, Paul Michel, and Graham Neubig. 2020. Weight poisoning attacks on pretrained models. In *Proceedings of ACL*.

Dianqi Li, Yizhe Zhang, Hao Peng, Liqun Chen, Chris Brockett, Ming-Ting Sun, and Bill Dolan. 2021. Contextualized perturbation for textual adversarial attack. In *Proceedings of NAACL*.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. BERT-ATTACK: Adversarial attack against BERT using BERT. In *Proceedings of EMNLP*.

Xiaojing Liao, Kan Yuan, XiaoFeng Wang, Zhongyu Pei, Hao Yang, Jianjun Chen, Haixin Duan, Kun Du, Eihal Alowaisheq, Sumayah Alrwais, et al. 2016.

Seeking nonsense, looking for trouble: Efficient promotional-infection detection through semantic inconsistency search. In *2016 IEEE Symposium on Security and Privacy (SP)*.

Hui Liu, Yongzheng Zhang, Yipeng Wang, Zheng Lin, and Yige Chen. 2020. Joint character-level word embedding and adversarial stability training to defend adversarial text. In *Proceedings of AAAI*.

Qingyuan Liu, Eduard C Dragut, Arjun Mukherjee, and Weiyi Meng. 2015. Florin: a system to support (near) real-time applications on user generated content on daily news. *Proceedings of the VLDB Endowment*.

Rishabh Maheshwary, Saket Maheshwary, and Vikram Pudi. 2021. Generating natural language attacks in a hard label black box setting. In *Proceedings of AAAI*.

Fiammetta Marulli, Laura Verde, and Lelio Campanile. 2021. Exploring data and model poisoning attacks to deep learning-based nlp systems. *Procedia Computer Science*.

Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of SIGIR*.

Vangelis Metsis, Ion Androutsopoulos, and Georgios Paliouras. 2006. Spam filtering with naive bayes-which naive bayes? In *CEAS*.

Maximilian Mozes, Pontus Stenetorp, Bennett Kleinberg, and Lewis Griffin. 2021. Frequency-guided word substitutions for detecting textual adversarial examples. In *Proceedings of ACL*.

Arjun Mukherjee, Vivek Venkataraman, Bing Liu, Natalie Glance, et al. 2013. Fake review detection: Classification and analysis of real and pseudo reviews. *UIC-CS-03-2013. Technical Report*.

Jan Neerbek. 2019a. EDENCE.

Jan Neerbek. 2019b. FAS.

Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of EMNLP-IJCNLP*.

Christina Niklaus, Matthias Cetto, André Freitas, and Siegfried Handschuh. 2018. A survey on open information extraction. *arXiv preprint arXiv:1806.05599*.

Xudong Pan, Mi Zhang, Shouling Ji, and Min Yang. 2020. Privacy risks of general-purpose language models. In *2020 IEEE Symposium on Security and Privacy (SP)*.

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

Nidhi A Patel and Rakesh Patel. 2018. A survey on fake review detection using machine learning techniques. In *2018 4th International Conference on Computing Communication and Automation (ICCCA)*.

Juan Carlos Pereira-Kohatsu, Lara Quijano-Sánchez, Federico Liberatore, and Miguel Camacho-Collados. 2019. Detecting and monitoring hate speech in twitter. *Sensors*.

Danish Pruthi, Bhuwan Dhingra, and Zachary C Lipton. 2019. Combating adversarial misspellings with robust word recognition. *Proceedings of ACL*.

Fanchao Qi, Yangyi Chen, Xurui Zhang, Mukai Li, Zhiyuan Liu, and Maosong Sun. 2021. Mind the style of text! adversarial and backdoor attacks based on text style transfer. In *Proceedings of EMNLP*.

Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. 2017. Truth of varying shades: Analyzing language in fake news and political fact-checking. In *Proceedings of EMNLP*.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of ACL*.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of ACL*.

Julian Risch and Ralf Krestel. 2020. Toxic comment detection in online discussions. In *Deep learning-based approaches for sentiment analysis*.

Joni Salminen, Chandrashekhar Kandpal, Ahmed Mohamed Kamel, Soon gyo Jung, and Bernard J. Jansen. 2022. Creating and detecting fake reviews of online products. *Journal of Retailing and Consumer Services*.

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

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*.

Chenglei Si, Zhengyan Zhang, Fanchao Qi, Zhiyuan Liu, Yasheng Wang, Qun Liu, and Maosong Sun. 2021. Better robustness by more coverage: Adversarial and mixup data augmentation for robust finetuning. In *Findings of ACL-IJCNLP*.

Chengai Sun, Qiaolin Du, and Gang Tian. 2016. Exploiting product related review features for fake review detection. *Mathematical Problems in Engineering*.

Samson Tan, Shafiq Joty, Lav R Varshney, and Min-Yen Kan. 2020. Mind your inflections! improving nlp for non-standard englishes with base-inflection encoding. *Proceedings of EMNLP*.

Stephen Tomlinson. 2010. Learning task experiments in the trec 2010 legal track. In *TREC*.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing NLP. In *Proceedings of EMNLP*.

Eric Wallace, Tony Zhao, Shi Feng, and Sameer Singh. 2021. Concealed data poisoning attacks on NLP models. In *Proceedings of NAACL*.

Boxin Wang, Hengzhi Pei, Boyuan Pan, Qian Chen, Shuohang Wang, and Bo Li. 2020a. T3: Tree-autoencoder constrained adversarial text generation for targeted attack. In *Proceedings of EMNLP*.

Peng Wang Wang, Xiaojing Liao Liao, Yue Qin, and XiaoFeng Wang. 2020b. Into the deep web: Understanding e-commercefraud from autonomous chat with cybercriminals. In *Proceedings of the ISOC Network and Distributed System Security Symposium (NDSS), 2020*.

Tianlu Wang, Xuezhi Wang, Yao Qin, Ben Packer, Kang Li, Jilin Chen, Alex Beutel, and Ed Chi. 2020c. CAT-gen: Improving robustness in NLP models via controlled adversarial text generation. In *Proceedings of EMNLP*.

Wenjie Wang, Pengfei Tang, Jian Lou, and Li Xiong. 2021a. Certified robustness to word substitution attack with differential privacy. In *Proceedings of NAACL*.

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

Xiaosen Wang, Yichen Yang, Yihe Deng, and Kun He. 2021c. Adversarial training with fast gradient projection method against synonym substitution based text attacks. *Proceedings of AAAI*.

Hajime Watanabe, Mondher Bouazizi, and Tomoaki Ohtsuki. 2018. Hate speech on twitter: A pragmatic approach to collect hateful and offensive expressions and perform hate speech detection. *IEEE access*.

Lei Xu, Yangyi Chen, Ganqu Cui, Hongcheng Gao, and Zhiyuan Liu. 2022. Exploring the universal vulnerability of prompt-based learning paradigm. In *Findings of NAACL*.

Fan Yang, Arjun Mukherjee, and Eduard Dragut. 2017. Satirical news detection and analysis using attention mechanism and linguistic features. *Proceedings of EMNLP*.

Mao Ye, Chengyue Gong, and Qiang Liu. 2020. SAFER: A structure-free approach for certified robustness to adversarial word substitutions. In *Proceedings of ACL*.

Kan Yuan, Haoran Lu, Xiaojing Liao, and XiaoFeng Wang. 2018. Reading thieves' cant: automatically identifying and understanding dark jargons from cybercrime marketplaces. In *27th {USENIX} Security Symposium ({USENIX} Security 18)*.

Kan Yuan, Di Tang, Xiaojing Liao, XiaoFeng Wang, Xuan Feng, Yi Chen, Menghan Sun, Haoran Lu, and Kehuan Zhang. 2019. Stealthy porn: Understanding real-world adversarial images for illicit online promotion. In *2019 IEEE Symposium on Security and Privacy (SP)*.

Lifan Yuan, Yichi Zhang, Yangyi Chen, and Wei Wei. 2021. Bridge the gap between cv and nlp! a gradient-based textual adversarial attack framework. *arXiv preprint arXiv:2110.15317*.

Yuan Zang, Bairu Hou, Fanchao Qi, Zhiyuan Liu, Xiaojun Meng, and Maosong Sun. 2020a. Learning to attack: Towards textual adversarial attacking in real-world situations. *arXiv preprint arXiv:2009.09192*.

Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun. 2020b. Word-level textual adversarial attacking as combinatorial optimization. In *Proceedings of ACL*.

Guoyang Zeng, Fanchao Qi, Qianrui Zhou, Tingji Zhang, Zixian Ma, Bairu Hou, Yuan Zang, Zhiyuan Liu, and Maosong Sun. 2021. OpenAttack: An open-source textual adversarial attack toolkit. In *Proceedings of ACL-IJCNLP*.

Lei Zhang, Shuai Wang, and Bing Liu. 2018. Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*.

Xichen Zhang and Ali A Ghorbani. 2020. An overview of online fake news: Characterization, detection, and discussion. *Information Processing & Management*.

Xinyi Zhou, Reza Zafarani, Kai Shu, and Huan Liu. 2019a. Fake news: Fundamental theories, detection strategies and challenges. In *Proceedings of WSDM*.

Yi Zhou, Xiaoqing Zheng, Cho-Jui Hsieh, Kai-Wei Chang, and Xuanjing Huang. 2021. Defense against synonym substitution-based adversarial attacks via Dirichlet neighborhood ensemble. In *Proceedings of ACL-IJCNLP*.

Yichao Zhou, Jyun-Yu Jiang, Kai-Wei Chang, and Wei Wang. 2019b. Learning to discriminate perturbations for blocking adversarial attacks in text classification. In *Proceedings of EMNLP-IJCNLP*.

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. 2020. Freelb: Enhanced adversarial training for natural language understanding. *Proceedings of ICLR*.

A Survey on Previous Work

We conduct a survey on previous adversarial attack methods about the specific tasks and datasets they employ in their evaluation. The results are listed in Table 7.

Work	Task	Dataset
(Alzantot et al., 2018)	SA; NLI.	IMDB; SNLI.
(Ren et al., 2019)	SA; NC; Topic classification.	IMDB; AG; Yahoo! Answers.
(Jin et al., 2020)	SA; NLI; NC; Fake News.	IMDB; Yelp; MR; SNLI; MNLI; AG; Fake News.
(Zang et al., 2020b)	SA; NLI.	IMDB; SST-2; SNLI.
(Li et al., 2020)	SA; NLI; NC; Fake News.	IMDB; Yelp; SNLI; MNLI; AG; Fake News.
(Maheshwary et al., 2021)	SA; NLI; NC.	IMDB; Yelp; MR; SNLI; MNLI; AG; Yahoo.
(Iyyer et al., 2018)	SA; NLI.	SST-2; SICK.
(Chen et al., 2021b)	SA; NLI; NC,	SST-2; MNLI; AG.
(Qi et al., 2021)	SA; NC; Hate-Speech.	SST-2; AG; Hate-Speech.
(Li et al., 2021)	SA; NLI; NC.	Yelp; MNLI; QNLI; AG.
(Li et al., 2021)	SA; NLI; NC.	Yelp; MNLI; QNLI; AG.
(Yuan et al., 2021)	SA; NLI; NC,	SST-2; MNLI; AG.

Table 7: Survey on previous work. **SA** stands for sentiment analysis. **NC** stands for news classification. Adversarial-oriented tasks and datasets are highlighted in **red**.

B Task Description

The task statistics are listed in Table 8. We give the task descriptions and our motivations to choose these tasks below.

B.1 Misinformation

Words in news media and political discourse have considerable power in shaping people's beliefs and opinions. As a result, their truthfulness is often compromised to maximize the impact on society (Zhang and Ghorbani, 2020; Zhou et al., 2019a; Fonseca et al., 2016). We generally believe that fake news is caused by objective factors such as misdeclarations, misdescriptions, or misuse of terminology. And this task is to detect misinformation that contains deceived or unverified information, including rumors, misreported, and satirical news.

B.2 Disinformation

In addition to misinformation caused by objective reasons, there is also a type of fake information caused by subjectively distorting facts. This type of information mainly concentrates on online comments and reviews in online shopping malls and online restaurant/hotel reservation websites to lure customers into consumption (Mukherjee et al., 2013; Sun et al., 2016; Patel and Patel, 2018). We define such a task as disinformation detection. In general, this task is dedicated to identifying deliberate fabrication of facts, including (1) Artificial comments reversing the black and white; (2) Generated nonexistent information.

B.3 Toxic

The rapid growth of information in social networks such as Facebook, Twitter, and blogs makes it challenging to monitor what is being published and spread on social media. Abusive comments are widespread on social networks, including cyberbullying, cyberterrorism, sexism, racism, and hate-speech. Thus, the primary objective of toxic detection is to identify toxic contents in the web, which is an essential ingredient for anti-bullying policies and protection of individual rights on social media (Pereira-Kohatsu et al., 2019; Bosco et al., 2018; Watanabe et al., 2018; Risch and Krestel, 2020).

B.4 Spam

In recent years, unwanted commercial bulk emails have become a huge problem on the internet. Spam emails prevent the user from making good use of time. More importantly, some spam emails contain fraud and phishing messages that can also cause financial damage to users (Fonseca et al., 2016). The Spam Classification Tasks is to detect spam information including scams, harassment, advertising, and promotion in Emails, SMS, and even chat messages to avoid unnecessary losses for users (Cormack et al., 2008).

B.5 Sensitive Information

Text documents shared across third parties or published publicly contain sensitive information by nature. Detecting sensitive information in unstructured data is crucial for preventing data leakage. This task is to detect sensitive information including intellectual property and product progress from companies, trading and strategic information of public institutions and organizations, and private information of individuals (Berardi et al., 2015; Chow et al., 2008; Grechanik et al., 2014).

C Definition of Validity

In general, the validity metric is to measure the preservation of adversarial meaning in the crafted adversarial samples. The adversarial meaning is task-specific and should be considered differently. So, the validity definition is relevant to the specific adversarial goal in the specific security task. In

Task	Dataset	Unbalanced				Balanced		
		|Train|	|Test|	Ave. Length	Ratio	|Train|	|Test|	Ave. Length
Misinformation	LUN	36148	15492	535.33	0.79	14906	6454	499.49
	SATNews	145677	37221	702.51	0.09	25264	7202	646.65
Disinformation	Amazon-LB	17902	8609	99.01	0.49	17434	8522	100.13
	CGFake	28290	12130	67.48	0.50	28290	12130	67.48
Toxic	HOSL	17348	7435	14.12	0.83	5832	2494	14.32
	Jigsaw2018	159560	63978	66.52	0.10	30587	12180	58.42
Spam	Enron	17774	7918	311.47	0.46	16159	7277	311.53
	SpamAssassin	3766	3774	291.75	0.28	2081	2066	308.50
Sensitive Information	EDENCE	105376	22577	22.45	0.24	51098	10328	21.79
	FAS	60470	16272	27.65	0.31	33814	13294	29.27

Table 8: Datasets statistics. The ratio refers to the proportion of fake/hate/spam/sensitive samples in corresponding datasets.

our Advbench, the adversarial meanings are exaggerated and satirical contents (Misinformation), inauthentic and untrue comments (Disinformation), abusive language (Toxic), illegal or time-wasting messages (Spam), and sensitive information embedded in common comments (Sensitive Information). So, the ultimate goal of attackers is to spread the adversarial meaning, no matter how many perturbations attackers introduce to other unrelated content.

D Real-world Adversarial Attack

We give some real-world adversarial cases collected from social media in Figure 1. Although these cases are written in Chinese, the perturbation rules are general and widely applicable. We can see that case-1, case-2, and case-5 also employ character-level perturbations, including substitution, deletion, and insertion. Besides, case-3 and case-4 employ the strategy of adding irrelevant and distracting words to the original sample. These samples can be easily comprehended by humans but easily fool the detection system. We employ these strategies in our simple method to simulate real-world adversarial attacks.

E Experimental Results

E.1 Attack Efficiency

Figure 2 shows the results of the attack success rate under the restriction of maximum query times.

E.2 Secondary Priority Metrics

We list the results of secondary priority metrics in Table 9.

E.3 Evaluation on the Defense Side

The results are shown in Table 10. We employ the SOTA defense method proposed in the NLP community (Mozes et al., 2021). This method identifies word substitutions by the frequency difference between the substituted word and its corresponding substituted word. The frequency distribution of words is obtained on the training set, and the detector is tuned on the validation set. Then, the detector can be employed to identify and restore adversarial samples in the inference time.

For each attack method, we input N adversarial samples (successfully attack the model) to the trained detector to obtain the number of samples detected as adversarial samples (n_{det}) and the number of samples successfully restored (n_{res}). Then the **detection rate** (R_{det}) and **restored rate** (R_{res}) are calculated according to the formula:

$$\begin{cases} R_{res} = \dfrac{n_{res}}{N} \\ R_{det} = \dfrac{n_{det}}{N} \end{cases} \tag{3}$$

F Experimental Details

For the sake of calculation speed and fairness, we truncate all sentences to the first 480 words. Then, we empirically set the hyper-parameters including distracting words, the insertion number of distracting words at the beginning and end of sentences, perturbation batch size, and perturbation epochs according to the attack performance and preservation of adversarial meaning. We only attack the original content in sentences, leaving out adversarial content introduced by our perturbations. The comprehensive settings of hyper-parameters are shown in Table 11.

Task	Disinformation			Toxic			Sensitive Information		
Method | Dataset	Amazon-LB			HSOL			EDENCE		
	Levenstein	ΔI	%PPL	Levenstein	ΔI	%PPL	Levenstein	ΔI	%PPL
TextFooler	25.68	0.57	**1.79**	10.83	0.03	0.89	11.86	0.10	2.51
PWWS	95.89	1.27	3.69	18.13	0.06	2.49	30.06	0.21	5.84
BERT-Attack	117.31	0.41	6.60	22.13	0.06	3.13	24.78	0.07	3.69
SememePSO(maxiter=100)	28.58	0.64	1.93	13.42	0.06	2.17	14.71	0.09	3.35
DeepWordBug(power=5)	**18.86**	0.33	4.19	**9.14**	-0.01	2.57	**6.66**	-0.02	5.13
DeepWordBug(power=25)	34.43	**0.23**	4.27	18.96	**-0.09**	1.32	19.57	**-0.04**	2.41
ROCKET	82.93	1.01	1.81	1083.42	44.99	**-0.98**	85.75	4.99	**-0.46**

Method | Dataset	CGFake			Jigsaw2018			FAS		
	Levenstein	ΔI	%PPL	Levenstein	ΔI	%PPL	Levenstein	ΔI	%PPL
TextFooler	16.62	0.23	2.12	14.64	0.09	1.21	13.12	0.05	2.45
PWWS	101.47	1.11	4.69	52.76	0.38	4.19	48.93	0.23	5.04
BERT-Attack	82.85	0.48	11.41	30.65	0.05	3.33	53.20	0.10	6.12
SememePSO(maxiter=100)	23.77	0.31	3.15	18.23	0.11	3.42	15.55	0.03	2.37
DeepWordBug(power=5)	**10.01**	0.05	5.16	**10.79**	0.03	3.45	**6.65**	-0.02	3.64
DeepWordBug(power=25)	29.49	**0.00**	4.13	20.49	**-0.05**	2.27	18.33	**-0.03**	2.44
ROCKET	38.27	1.03	**1.66**	1084.44	44.99	**-0.96**	97.95	5.01	**0.34**

Table 9: Results of secondary priority metrics considering perturbation degree and grammaticality.

Task	Disinformation		Toxic		Sensitive Information	
Method | Dataset	Amazon-LB		HSOL		EDENCE	
	$R_{res}(\%)$	$R_{det}(\%)$	$R_{res}(\%)$	$R_{det}(\%)$	$R_{res}(\%)$	$R_{det}(\%)$
TextFooler	57.89	73.68	85.00	86.00	43.40	53.19
PWWS	49.28	69.57	87.76	89.80	46.62	58.82
BERT-Attack	18.29	43.90	17.02	33.51	25.31	33.93
SememePSO(maxiter=100)	52.54	79.66	79.70	83.16	60.28	66.92
DeepWordBug(power=5)	**0.00**	**10.00**	6.05	12.46	4.41	7.05
DeepWordBug(power=25)	1.49	11.94	**0.94**	**1.41**	**0.88**	**1.88**
ROCKET	14.15	35.05	12.75	30.03	8.11	15.75

Method | Dataset	CGFake		Jigsaw2018		FAS	
	$R_{res}(\%)$	$R_{det}(\%)$	$R_{res}(\%)$	$R_{det}(\%)$	$R_{res}(\%)$	$R_{det}(\%)$
TextFooler	4.42	7.73	22.58	33.06	56.90	63.79
PWWS	5.10	7.43	68.37	70.92	61.92	71.23
BERT-Attack	**0.11**	6.37	15.26	22.37	27.89	37.91
SememePSO(maxiter=100)	9.43	13.73	50.77	61.20	71.29	75.37
DeepWordBug(power=5)	1.69	13.08	1.67	12.53	1.23	7.35
DeepWordBug(power=25)	1.17	8.45	**0.35**	**3.47**	**0.13**	**1.80**
ROCKET	0.31	**2.49**	7.00	22.26	5.64	13.11

Table 10: Results on the defense side.

Parameter | Task	Misinformation	Disinformation	Toxic	Spam	Sensitive Information
Distracting Word	reuters	up	peace	>	any
Prefix Number	5	3	0	10	0
Postfix Number	30	8	180	30	20
Batch Size	8	6	4	6	3
Epoch	3	3	3	2	3
Max Modification Number	100	100	180	30	100

Table 11: Hyper-parameters of ROCKET on each task.

Figure 1: Real-world cases of adversarial attacks. Adversarially modified content is highlighted in **red**.

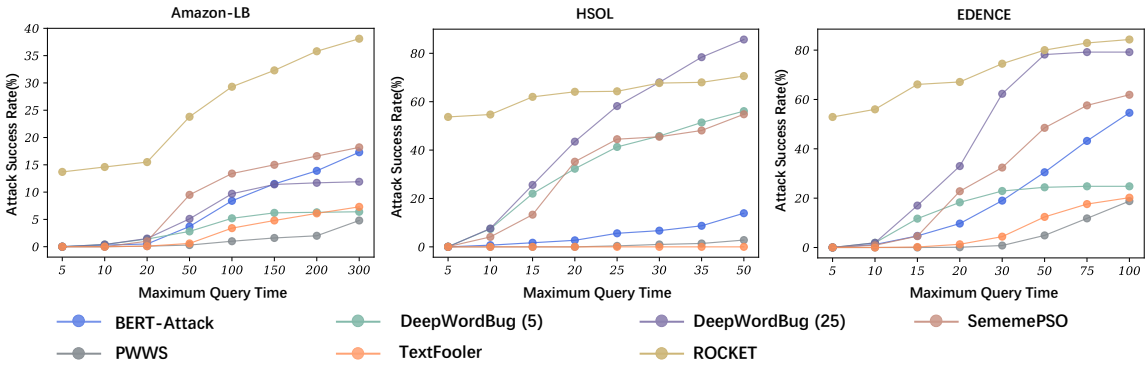

Figure 2: Attack success rate under the restriction of maximum query times.

Here we give our intuition for choosing distracting words for each task. For misinformation detection, we find that newspaper names often appear at the beginning or end of the news. So, we insert a few "Reuters" before and after the sentence without affecting the validity of the main content. For Disinformation detection, adding several encouraging words "up" does not affect the judgment of the authenticity of the comments, so we use a number of "up" as parenthetical words. For toxic detection, we need to use friendly and harmonious words to fool detectors. So, we insert many "peace" to the sentences. For spam detection, we find that ">" sometimes appears in emails to separate the text. So, we use a large number of them as inserted words, which doesn't affect the nature of the original sentence. For sensitive information detection, we employ "any" as we find that samples that are often classified as non-sensitive contain adverbs at the end.

G Human Evaluation Details

We set up a human evaluation to further evaluate the validity of adversarial samples. We choose the disinformation and toxic detection tasks because the validity definitions are clear and can be easily understood by annotators. For each task, we consider 2 corresponding datasets and sample 100 original and adversarial samples pairs for each attack method. For each pair, we ask 3 human annotators to evaluate whether the adversarial meaning is preserved in the adversarially crafted sample (validity). They need to give a validity score from 0-2 for each pair, where 2 means that the adversarial meaning has been perfectly preserved, 1 means that the sentence meaning is ambiguous but may still preserve some adversarial meaning, and 0 means that the crafted adversarial sample don't preserve any adversarial meaning in the original sample. We use the voting strategy to produce the annotation results of validity for each adversarial sample. Then we average the scores for all 100 samples in each task as the final validity score for each attack method. The results are shown in Table 6.