On the Universal Adversarial Perturbations for Efficient Data-free Adversarial Detection

Songyang Gao[1], **Shihan Dou**[1], **Qi Zhang**[12]*, **Xuanjing Huang**[12], **Jin Ma**[3], **Ying Shan**[3]
[1] School of Computer Science, Fudan University, Shanghai, China
[2] Shanghai Key Laboratory of Intelligent Information Processing, Shanghai, China
[3]Tencent PCG
{gaosy21, shdou21}@m.fudan.edu.cn

Abstract

Detecting adversarial samples that are carefully crafted to fool the model is a critical step to socially-secure applications. However, existing adversarial detection methods require access to sufficient training data, which brings noteworthy concerns regarding privacy leakage and generalizability. In this work, we validate that the adversarial sample generated by attack algorithms is strongly related to a specific vector in the high-dimensional inputs. Such vectors, namely UAPs (Universal Adversarial Perturbations), can be calculated without original training data. Based on this discovery, we propose a data-agnostic adversarial detection framework, which induces different responses between normal and adversarial samples to UAPs. Experimental results show that our method achieves competitive detection performance on various text classification tasks, and maintains an equivalent time consumption to normal inference.

1 Introduction

Despite remarkable performance on various NLP tasks, pre-trained language models (PrLMs), like BERT (Devlin et al., 2018), are highly vulnerable to adversarial samples (Zhang et al., 2020; Zeng et al., 2021). Through intentionally designed perturbations, attackers can modify the model predictions to a specified output while maintaining syntactic and grammatical consistency (Jin et al., 2020; Li et al., 2020b). Such sensitivity and vulnerability induce persistent concerns about the security of NLP systems (Zhang et al., 2021c). Compared to deploying robust new models, it would be more applicable to production scenarios by distinguishing adversarial examples from normal inputs and discarding them before the inference phase (Shafahi et al., 2019). Such detection-discard strategy helps to reduce the effectiveness of adversarial samples and can be combined with existing defence methods (Mozes et al., 2021).

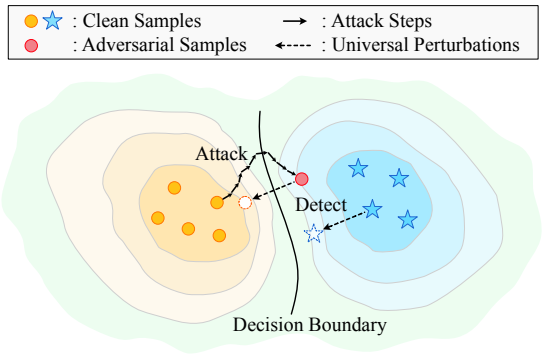

Figure 1: Illustration of our UAPAD framework. The solid and hollow markers represent samples before and after the universal perturbation. The adversarial samples are embedded closer to the decision boundary to maintain similarity with the original samples, resulting in differential resistance to universal adversarial perturbations (UAPs) with clean samples. We construct our detection framework based on this observation.

However, existing adversarial detection methods depend heavily on the statistical characteristics of the training data manifolds, such as density estimation (Yoo et al., 2022) and local intrinsic dimensionality (Liu et al., 2022). Some other researches focus on identifying high-frequency words in the training data and replacing or masking them in the prediction phase to observe the change in logits score (Mozes et al., 2021; Mosca et al., 2022). We propose a summary of existing works in Table 1. All these detection methods assume that training data is available, which suffers from the following two problems: (1) Some companies only provide model checkpoints without customer data due to privacy and security issues. (2) Some datasets can be large so it is not practical or convenient to save and process them on different platforms.

In this work, we propose UAPAD, a novel framework to detect adversarial samples without exposure to training data and maintain a time consumption consistent with normal inference. We visualize our detection framework in Figure 1. Universal adversarial perturbations (UAPs) is an intriguing

* Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2023, pages 13573–13581
July 9-14, 2023

Method	Summary	Require Clean Data	Require Adv. Data	Require Extra Model
MLE (Lee et al., 2018)	Gaussian discriminant analysis	✔	✖	✔
DISP (Zhou et al., 2019)	Token-level detection model	✔	✔	✔
FGWS (Mozes et al., 2021)	Frequency-based word substitution	✔	✔	✖
ADFAR (Bao et al., 2021)	Sentence-level detection model	✔	✔	✔
RDE (Yoo et al., 2022)	Feature-based density estimation	✔	✖	✖
UAPAD (Ours)	Universal adversarial perturbation	✖	✖	✖

Table 1: Summary of previous detection methods in NLP system. Requiring clean/adv. data indicates what data is needed for the training and validation process. Requiring extra models indicates whether a separate new model needs to be trained for adversarial detection. Our approach is data-agnostic and can be easily integrated into the inference phase.

phenomenon on neural models, i.e. a single perturbation that is capable to fool a DNN for most natural samples (Zhang et al., 2021b), and can be calculated without the original training data (Moposuri et al., 2018; Zhang et al., 2021a). We explore the utilization of UAPs to detect adversarial attacks, where adversarial and clean samples exhibit differential resistance to pre-trained perturbations on a sensitive feature subspace.

Experimental results demonstrate that our training-data-agnostic method achieves promising detection accuracy with BERT on multiple adversarial detection tasks without using training or adversarial data, consuming additional inference time, or conducting overly extensive searches for hyperparameters. Our main contributions are as follows:

- We analyze and verify the association between adversarial samples and an intrinsic property of the model, namely UAPs, to provide a new perspective on the effects of adversarial samples on language models.
- We propose a novel framework (**UAPAD**), which efficiently discriminates adversarial samples without access to training data, and maintains an equivalent time consumption to normal inference. Our *codes*[1] are publicly available.

2 Related Work

2.1 Universal adversarial perturbation

The existence of UAPs has first been demonstrated by (Moosavi-Dezfooli et al., 2017), that a single perturbation can fool deep models when added to most natural samples. Such phenomena have been extensively verified in image (Khrulkov and Oseledets, 2018), text (Song et al., 2021), and audio models (Li et al., 2020a). Some works attribute the existence of UAPs to a specific low-dimensional subspace, which is perpendicular to the decision boundary for most of the data. The attention on UAPs mainly focused on their construction, detection and defence (Zhang et al., 2021b), and neglected to explore the relationship between adversarial samples and UAPs. Our experimental results in Figure 2 demonstrate the tight connection between these two phenomena.

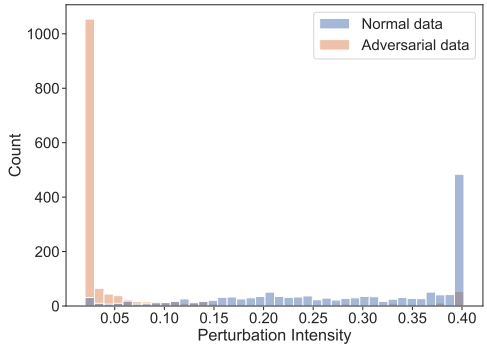

Figure 2: Illustration of the different resistance to universal perturbations in adv and clean data. Predictions for adversarial samples are inverted by a small perturbation intensity while clean samples maintain the original results.

2.2 Adversarial detection in NLP

Adversarial detection is an emerging area of research on language model security. A series of works analyze the frequency characteristics of word substitutions in pre-collected adversarial sentences and replace (Zhou et al., 2019; Mozes et al., 2021) or mask (Mosca et al., 2022) them to observe model reactions. These methods rely on empirically designed word-level perturbations, which limit their generalizability across different attacks. Ma et al. (2018) first proposed to train additional

[1] https://github.com/SleepThroughDifficulties/UAPAD.git

discriminative models to decide whether an input sentence has suffered from word-level adversarial substitution. This idea was generalized by Liu et al. (2022) and Yoo et al. (2022), which determine the likelihood of a sentence has been perturbed. However, they still require the statistical characteristics of the training data. In this paper, we for the first time propose to construct data-agnostic models and achieve remarkable detection results.

3 Method

This section shows how to calculate the UAPs for a specific text model without obtaining training data. And subsequently, how to detect adversarial data by pre-trained UAPs.

Data-free UAPs We compute UAPs for a fine-tuned model by perturbing the substitute inputs, based on the fact that UAPs are generalized properties for a given model. We start with a parameter-frozen target network f and a random perturbation δ. The optimal situation is we can obtain some data that are involved in the training procedure. However, there are situations that we cannot access to training samples or it is unclear whether the accessible data is within the training set. To demonstrate the effectiveness of UAPAD under the data-agnostic scenario. We initialize the input embedding by randomly selecting data from an unrelated substitute dataset (e.g., the MNLI dataset in our experiments). It is a reasonable assumption that a defender can access a moderate amount of substitute data. These embeddings are subsequently updated to ensure the model's confidence score is above the threshold on them. In our framework, we only retain samples with model confidence above 85% to calculate UAPs. We then optimize the perturbation δ by gradient-ascending the overall loss when added to all the inputs and project it to a normalized sphere of fixed radius to constrain its norm. We obtain a reasonable UAP when most predictions are induced to a fixed result under perturbation.

Adversarial Detection with UAPs In Figure 2, we illustrate the different resistance to UAPs between clean and adversarial samples. We utilize this property to conduct adversarial detection. Given an input x, we perform one inference on model f to obtain the normal output $y = f(x)$ and perform another one when x is perturbed by a calculated UAP δ, that is $y' = f(x + w * \delta)$, where w is a hyperparameter controlling the perturbation's intensity. We detect the input as an adversarial sample when $y \neq y'$. Noting that these two inferences can be computed in parallel, our approach does not introduce growth in inference time.

4 Experimental Setup

We experimentally validate our method on three well-accepted benchmarks: SST-2 (Socher et al., 2013), IMDB (Maas et al., 2011), and AGNews (Zhang et al., 2015). The statistics of involved benchmark datasets are summarised in Appendix A. We use the BERT-base (Devlin et al., 2018) as the target model and pre-generate adversarial samples for the detection task with three attack methods: TextFooler (Jin et al., 2020), PWWS (Ren et al., 2019), and BERTAttack (Li et al., 2020b).

4.1 Detecting Scenarios

Adversarial detection task requires a dataset $\mathcal{D}$, containing both clean samples $\mathcal{D}_{clean}$ and adversarial samples $\mathcal{D}_{adv}$. In the previous works, there exist two different strategies to construct adversarial datasets. Scenario 1 (easy): The adversarial dataset consists of only successful attack samples. Scenario 2 (hard): The adversarial dataset contains both successful and unsuccessful attack samples. Scenario 2 presents more challenging requirements for detection methods and is closer to real-world settings. We conduct experiments in both scenarios to fully illustrate the performance of UAPAD.

4.2 Implementation Details

We fine-tuned BERT using consistent settings with (Devlin et al., 2018). For all three datasets, we took 1500 training samples and saved their attack results under different attack algorithms as adversarial samples. UAPAD has a single hyperparameter w (strength of universal perturbation), which we set to 0.5 for all our detection experiments. Although we believe that a better weight exists and can boost the detection performance, we refuse to extend hyper-parameter searching which is against our original purpose. More implementation details and hyperparameters can be found in Appendix B.

Evaluation Metrics We use two metrics to measure our experimental results. **Detection accuracy (ACC)** measures the accuracy of classification results on all samples, and **F1-score (F1)** measures the harmonic mean of precision and recall scores. Similar to DISP, our method provides a direct dis-

Datasets	Methods	TextFooler		PWWS		BERT-ATTACK	
		Acc	F1	Acc	F1	Acc	F1
SST-2	MLE	79.6	77.0	77.5	77.2	70.3	63.7
	DISP	71.2	66.0	74.4	70.9	70.8	65.4
	FGWS	70.2	63.5	82.5	**81.3**	70.3	63.7
	RDE	78.0	73.4	79.5	77.6	83.4	81.3
	UAP (Ours)	**83.7**	**83.1**	**82.5**	80.9	**87.4**	**87.6**
IMDB	MLE	83.7	**81.8**	**81.5**	**79.4**	83.7	**82.3**
	DISP	68.8	70.6	66.8	68.2	67.3	68.8
	FGWS	74.7	69.7	77.5	74.0	74.4	69.3
	RDE	83.2	75.6	82.0	74.4	83.5	76.6
	UAP (Ours)	**84.1**	72.6	81.2	73.3	**83.8**	78.4
AGNEWS	MLE	79.9	79.6	77.3	76.9	82.7	78.6
	DISP	86.7	86.4	86.9	86.6	83.5	82.6
	FGWS	68.3	59.6	75.0	70.6	68.2	59.4
	RDE	85.0	86.7	85.8	81.4	88.2	88.0
	UAP (Ours)	**95.8**	**95.6**	**94.4**	**93.1**	**94.9**	**94.9**

Table 2: Adversarial detection results on easy scenario.

Datasets	Methods	TextFooler		PWWS		BERT-ATTACK	
		Acc	F1	Acc	F1	Acc	F1
SST-2	MLE	76.7	75.2	77.2	77.9	82.4	83.0
	DISP	71.2	64.9	74.4	68.8	70.7	64.9
	FGWS	65.7	55.6	69.3	61.7	64.6	53.5
	RDE	77.4	73.2	78.6	77.7	82.9	81.0
	UAP (Ours)	**80.7**	**80.9**	**78.8**	**78.1**	**84.9**	**85.1**
IMDB	MLE	74.0	**82.4**	74.5	76.2	74.4	78.9
	DISP	64.1	53.7	62.4	51.9	63.2	52.6
	FGWS	62.1	47.4	63.5	49.8	58.6	39.6
	RDE	77.4	73.2	**78.6**	77.7	**82.9**	**81.3**
	UAP (Ours)	**78.3**	76.8	77.5	**76.8**	79.3	78.0
AGNEWS	MLE	77.9	77.0	73.2	71.5	79.2	78.9
	DISP	86.1	84.5	85.4	81.0	83.1	81.5
	FGWS	64.7	52.8	67.7	58.4	64.1	51.6
	RDE	85.1	84.4	77.0	78.5	86.6	85.7
	UAP (Ours)	**88.6**	**88.1**	**86.3**	**81.4**	**92.0**	**91.9**

Table 3: Adversarial detection results on hard scenario.

criminant rather than a score and therefore does not apply to the AUC metric.

Baselines We compare our proposed methods with four strong baselines. Details are summarized in Appendix C.

- **MLE** (Lee et al., 2018) proposes to train detection models based on Mahalanobis distance.
- **DISP** (Zhou et al., 2019) verifies the likelihood that a token has been perturbed.
- **FGWS** (Mozes et al., 2021) substitutes low-frequency words in the sentence to detect Word-level attacks.
- **RDE** (Yoo et al., 2022) models the probability density of inputs and generates the likelihood of a sentence being perturbed.

5 Experiment Results and Discussions

In this section, we show the experimental performance of our proposed method under the two scenarios in Section 4.1, and investigate different defence methods on the inference time consumption.

5.1 Main Results

Table 2 and 3 show the detect results on three datasets and three attacks. The highest means are marked in **bold**. Out of the 18 combinations of dataset-attack-scenario, UAPAD achieves the best performance on 15 of them on ACC and 12 of them on F1 metric, which demonstrates the competitiveness of our data-agnostic approach. UAPAD guarantees remarkable detection performance on the SST-2 and AGNews datasets and suffers from a small degradation on the IMDB dataset. We argue that the average length of sentences is greater on IMDB, resulting in stronger dissimilarity between the adversarial sample generation by attack algorithms and the original sentence. On the AGNews dataset, UAPAD provided a 3-11% increase in detection accuracy relative to the baseline approach. We attribute this impressive improvement to more categories on this task, which improved the accuracy of estimation on the model's UAPs.

5.2 Time Consumption

To further reveal the strength of UAPAD besides its detection performance, we compare its GPU training time consumption with other baseline methods. As is demonstrated in Table 4, The time consumption of UAPAD is superior to all the comparison methods. Only FGWS (Mozes et al., 2021) exhibits similar efficiency to ours (with about 20% time growth on SST-2 and IMDB). FGWS neither contains a backpropagation process in the inference phase, but still requires searching the pre-built word list for substitution.

Methods	**SST-2**	IMDB
finetune	3.42	4.33
UAP (Ours)	**3.58**	**4.61**
DISP	19.2	24.6
FGWS	4.17	5.58
RDE	76.8	102.3

Table 4: GPU time consumption (seconds) of detection 1500 samples. UAPAD costs nearly the same as normal predictions.

6 Conclusion

In this paper, we propose that adversarial samples and clean samples exhibit different resistance to UAPs, a model-related vector that can be calculated without accessing any training data. Based

on this discovery, we propose UAPAD as an efficient and application-friendly algorithm to overcome the drawbacks of previous adversarial detection methods in terms of slow inference and the requirement of training samples. UAPAD acts by observing the feedback of inputs when perturbed by pre-computed UAPs. Our approach achieves impressive detection performance against different textual adversarial attacks in various NLP tasks. We call for further exploration of the connection between adversarial samples and UAPs.

Acknowledgements

The authors wish to thank the anonymous reviewers for their helpful comments. This work was partially funded by National Natural Science Foundation of China (No.61976056,62076069) and Natural Science Foundation of Shanghai (23ZR1403500).

7 Limitations

This section discusses the potential limitations of our work. This paper's analysis of model effects mainly focuses on common benchmarks for adversarial detection, which may introduce confounding factors that affect the stability of our framework. Our model's performance on more tasks and more attack algorithms is worth further exploring. Our detection framework exploits the special properties exhibited by the adversarial sample under universal perturbation. We expect a more profound exploration of improving the connection between UAPs and adversarial samples. In Figure 2, we note that a small number (about 3%) of clean and adversarial samples do not suffer from UAP interference. It is worth conducting an analysis of them to further explore the robustness properties of the language models. We leave these problems to further work.

References

Rongzhou Bao, Jiayi Wang, and Hai Zhao. 2021. Defending pre-trained language models from adversarial word substitution without performance sacrifice. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3248–3258.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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

Valentin Khrulkov and Ivan Oseledets. 2018. Art of singular vectors and universal adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8562–8570.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. 2018. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems*, 31.

Jiguo Li, Xinfeng Zhang, Chuanmin Jia, Jizheng Xu, Li Zhang, Yue Wang, Siwei Ma, and Wen Gao. 2020a. Universal adversarial perturbations generative network for speaker recognition. In *2020 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE Computer Society.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020b. Bert-attack: Adversarial attack against bert using bert. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6193–6202.

Na Liu, Mark Dras, and Wei Emma Zhang. 2022. Detecting textual adversarial examples based on distributional characteristics of data representations. In *Proceedings of the 7th Workshop on Representation Learning for NLP*, pages 78–90.

Xingjun Ma, Bo Li, Yisen Wang, Sarah M Erfani, Sudanthi Wijewickrema, Grant Schoenebeck, Dawn Song, Michael E Houle, and James Bailey. 2018. Characterizing adversarial subspaces using local intrinsic dimensionality. In *International Conference on Learning Representations*.

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

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. 2017. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1765–1773.

Konda Reddy Mopuri, Aditya Ganeshan, and R Venkatesh Babu. 2018. Generalizable data-free objective for crafting universal adversarial perturbations. *IEEE transactions on pattern analysis and machine intelligence*, 41(10):2452–2465.

Edoardo Mosca, Shreyash Agarwal, Javier Rando Ramírez, and Georg Groh. 2022. "that is a suspicious reaction!": Interpreting logits variation to detect nlp adversarial attacks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7806–7816.

Maximilian Mozes, Pontus Stenetorp, Bennett Kleinberg, and Lewis Griffin. 2021. Frequency-guided word substitutions for detecting textual adversarial examples. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 171–186.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 1085–1097.

Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. 2019. Adversarial training for free! *Advances in Neural Information Processing Systems*, 32.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Liwei Song, Xinwei Yu, Hsuan-Tung Peng, and Karthik Narasimhan. 2021. Universal adversarial attacks with natural triggers for text classification. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3724–3733.

KiYoon Yoo, Jangho Kim, Jiho Jang, and Nojun Kwak. 2022. Detection of adversarial examples in text classification: Benchmark and baseline via robust density estimation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3656–3672.

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 the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 363–371.

Chaoning Zhang, Philipp Benz, Adil Karjauv, Jae Won Cho, and In So Kweon. 2021a. Towards datafree universal adversarial perturbations with artificial images. In *RobustML workshop at ICLR2021*.

Chaoning Zhang, Philipp Benz, Chenguo Lin, Adil Karjauv, Jing Wu, and In So Kweon. 2021b. A survey on universal adversarial attack. *arXiv preprint arXiv:2103.01498*.

Wei Emma Zhang, Quan Z Sheng, Ahoud Alhazmi, and Chenliang Li. 2020. Adversarial attacks on deep-learning models in natural language processing: a survey. *ACM Transactions on Intelligent Systems and Technology*, 11(3):1–41.

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

Xinze Zhang, Junzhe Zhang, Zhenhua Chen, and Kun He. 2021c. Crafting adversarial examples for neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1967–1977.

Yichao Zhou, Jyun-Yu Jiang, Kai-Wei Chang, and Wei Wang. 2019. Learning to discriminate perturbations for blocking adversarial attacks in text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4904–4913.

A Dataset Statistics

Dataset	Train/Test	Classes	#Words
SST-2	67k/1.8k	2	19
IMDB	25k/25k	2	268
AGNews	120k/7.6k	4	40

Table 5: Statistics of datasets. In our experiments, we partition an additional 10 percent of the training set as the validation set to calculate the DSRM of the model.

B Experimental Details

In this appendix, we show the hyper-parameters used for our proposed method. We fine-tune the BERT-base model by the official default settings. For SST-2, we use the official validation set, while for IMDB and AGNews, we use 10% of the data in the training set as the validation set. The validation set and the adversarial samples generated using the validation set are used to select hyper-parameters. All three attacks are implemented using TextAttack[2] with the default parameter settings. Following Zhou et al. (2019), for SST-2, IMDB and AGNews, we build a balanced set consisting of 1500 clean test samples and 1500 adversarial samples to evaluate our proposed methods and all the baselines in this paper. We train our models on NVIDIA RTX 3090 GPUs (four for RDE and one for other methods). All experiments are run on three different seeds and report the mean result.

C Baseline Details

We compare our proposed detectors with three strong baselines in adversarial example detection.

MLE (Lee et al., 2018): A simple yet effective method for detecting OOD and adversarial examples in the image processing domain. The main idea is to induce a generative classifier under Gaussian discriminant analysis, resulting in a detection score based on Mahalanobis distance.

DISP (Zhou et al., 2019): A novel BERT-based framework can identify perturbations and correct malicious perturbations. it contains two independent components, a perturbation discriminator and an estimator for token recovery. To detect adversarial attacks, the discriminator verifies the likelihood that a token in the sample has been perturbed.

FGWS (Mozes et al., 2021) leverages the frequency properties of adversarial word substitution for the detection of adversarial samples. Briefly, FGWS replaces the low-frequency words with their most frequent synonyms in the dictionary to detect the perturbation.

RDE (Yoo et al., 2022) proposes a competitive adversarial detector based on density estimation. RDE models the probability density of the entire text and generates the likelihood of a text being perturbed.

[2]https://github.com/QData/TextAttack

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section 7: limitations

☑ A2. Did you discuss any potential risks of your work?
Section 7: limitations

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 4: experimental setup

☑ B1. Did you cite the creators of artifacts you used?
Section 4: experimental setup

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Section 4: experimental setup

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

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

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
In the appendix.A

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
In the appendix.A

C ☑ Did you run computational experiments?

Section 5: Experiment Results and Discussions

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 4: experimental setup

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

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 5: Experiment Results and Discussions

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 5: Experiment Results and Discussions

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 5: Experiment Results and Discussions

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

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.