Word-Level Adversarial Defense Layer for Robust Natural Language Classification

Mert Erkul and **Noè Canevascini** and **Maximilian Herde** and **Yannick Schnider**
ETH Zurich, D-INFK
{merkul, noec, herdem, yannicks}@ethz.ch

Abstract

Deep Neural Networks (DNNs) are frequently used in NLP for various tasks, such as classification and machine translation. However, recent results show that they are prone to adversarial attacks. Typically, defense strategies against the attacks either augment the training dataset, modify the word embeddings or use model-dependent algorithms. Such techniques suffer from transferability issues and heavy computations. In this study, we propose a computationally efficient, model and attack agnostic algorithm called Word-Level Adversarial Defense Layer (WLADL), which has been evaluated on text classification tasks with different architectures. For comparison, we applied the vanilla adversarial training (VAT) strategy (augmenting the dataset with successful adversarial examples), and the Synonym Encoding Method (SEM) to generate new word embeddings. We evaluate the defense strategies through their clean test results, alterations in the accuracies and adversarial query counts compared to non-defended models when attacked. Our experiments demonstrate that, when compared with VAT and SEM, WLADL shows competitive performance, while being a transferable algorithm that does not require any precomputations.

1 Introduction

In recent years, deep learning models gained significant popularity because of their remarkable performances in numerous tasks. However, these models have shown to be vulnerable against adversarial perturbations: minimal changes that are unperceivable by a human observer. Such perturbations fool the models to make false predictions (Goodfellow et al., 2015). In the NLP domain, existing adversarial attacks can be roughly divided in four categories: Character-level attacks consist of intentional typos (Ebrahimi et al., 2018; Gao et al., 2018), word-level attacks comprise low-frequency synonym replacement (Samanta and Mehta, 2017) for classification or antonym replacement for machine translation tasks (Cheng et al., 2019), sentence-level attacks change word positions (Zhang et al., 2019) and multi-level attacks add words to sentences for gradient disturbance (Song et al., 2021). In this study, we focus on examining defense strategies on word-level adversarial attacks for document classification tasks. Examples for word-level adversarial attacks can be seen in Table 1.

Several problems arise while applying VAT (Goodfellow et al., 2015) or more advanced defense algorithms (Wang et al., 2021b,a). To begin with, the VAT pipeline consists of training a clean model, attacking it using a designated attack strategy, selecting successful adversarial examples and retraining the model by augmenting the training dataset with the successful adversarial examples. It is necessary to select a candidate model and an attack to complete the pipeline; one can suggest that the selected attack might prove successful to use for one model but completely fail for another one. Also, training the model with the adversarial examples generated through a single attack might not improve performance for a different type of attack. Furthermore, complex defense algorithms such as the SEM (Wang et al., 2021b) also suffer from transferability and computational burden. SEM creates new word embeddings that group semantically similar words together, to avoid adversarial attacks that substitute words with rarely used synonyms. With SEM, BERT-like models are not able to fully utilize contextualized embeddings (Devlin et al., 2019).

We therefore introduce a modular, attack and model agnostic defense strategy: the Word-level Adversarial Defense Layer (WLADL) which works similar to a dropout layer. Compared to other defense strategies, WLADL does not require any precomputations, and is a training-time algorithm that can be easily applied to various types of model architectures.

Table 1: Adversarial examples and their predictive outputs for all datasets generated with Bidirectional LSTM attacked by PWWS

Dataset	Original Text	Adversarial Example	Ground Truth	Predicted Output	Perturbed Output
IMDb	A very comical but down to earth look into the behind the scene workings of an Australian bowling club. The way they deal with various problems such as takeovers, memberships and general running of the club, not to mention the car parking dilemma was well scripted.	A very comical but down to earth look into the behind the scene workings of an Australian bowling club. The way they deal with various problems such as takeovers, memberships and general running of the club, not to mention the car parking dilemma was swell scripted.	Positive	Positive (98%)	Negative (80%)
Yahoo! Answers	What's the best way to fight a cold? Take zinc or try Zycam homopathic remedy at any drug store or grocery.	What's the best way to fight a cold? Take zinc or try Zycam homopathic amend at any drug store or grocery.	Health	Health (46%)	Education (62%)
AG News	Sneaky Credit Card Tactics. Keep an eye on your credit card issuers. They may be about to raise your rates.	Sneaky Credit Card Tactics. Keep an eye on your credit card issuers. They may be about to kindle your rates.	Business	Business (78%)	Science/Technology (83%)

2 Models and Methods

To assess the performance of WLADL, we used the following baseline adversarial defense strategies, models and datasets from the literature.

2.1 Classifiers and Datasets

We selected three datasets and classifiers which are widely used as benchmarks in adversarial NLP literature. The primary focus for the model decisions was testing performance on different architectural designs. Therefore, we chose **Bidirectional LSTM (BiLSTM)** (recurrent), **Convolutional Neural Networks (CNN)** (convolutional) and fine-tuned **BERT** (transformer) (Devlin et al., 2019) as our base classifiers. For the datasets, we focused on increased variety in the document length, dataset size and number of classes. The three datasets that satisfy these requirements are **IMDb** (Maas et al., 2011), **AG News** (Zhang et al., 2015) and **Yahoo! Answers** (Zhang et al., 2015).

2.2 Baseline Defense Strategies

We chose VAT as the initial baseline defense algorithm and generated examples using a BiLSTM model trained on each of the clean datasets. Then, we attacked the model using Probability Weighted Word Saliency (Ren et al., 2019) (PWWS), generating approximately 10% adversarial samples for the IMDb and AG News training sets and as many examples as possible in 24 hours for Yahoo! Answers, to be computationally comparable to WLADL.

Another baseline we selected is SEM (Wang et al., 2021b), to compare WLADL with a strategy that modifies the word embeddings. It reduces an existing embedding matrix by mapping *similar* words to the most used one. We used Euclidean distance of the word embeddings for measuring similarity. The hyperparameters are the minimum euclidean distance (δ) to be seen as synonyms and the maximal number of synonyms (k) which can be mapped to the same word. Looking at the performance of our datasets and following the authors $\delta = 3.1$ and $k = 10$ were selected. We generated the new embedding matrices for every dataset using the most frequent 50k tokens.

2.3 Attacks

In literature, adversarial attacks are separated into white-box and black-box attacks. Black-box attacks do not make use of information regarding model parameters, whereas white-box attacks can also utilize them to perturb samples (Garg and Ramakrishnan, 2020). For this study, we opted for black-box attacks. The objective for a black-box word-level adversarial attack is as follows: Given a tokenized input, $X_i = [x_1^i, x_2^i, ..., x_n^i]$, a trained classifier C, and an output class y_i, an adversary searches for the *minimally necessary perturbations* of the tokens x_j^i yielding X_i^{adv}, such that $C(X_i) = y_i$ while $C(X_i^{adv}) \neq y_i$ (Garg and Ramakrishnan, 2020; Alzantot et al., 2018). Adversaries can employ external sources such as language-models, a WordNet (Fellbaum, 1998) thesaurus and embeddings such as GloVe (Pennington et al., 2014) to execute the attacks, but they have to respect certain constraints. Examples of such constraints are: the maximal number of queries, maximal number of perturbations in a document and grammatical correctness (Ren et al., 2019; Garg and Ramakrishnan, 2020; Alzantot et al., 2018). For the attacks that utilize external resources we

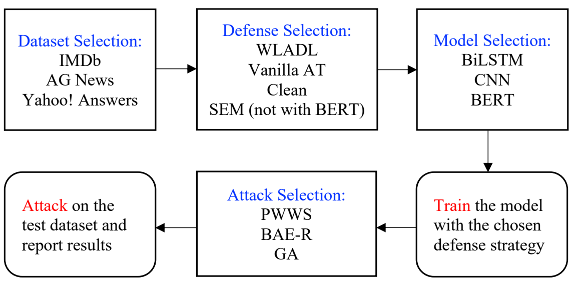

Figure 1: Experimental Pipeline

Algorithm 1: Word-Level Adversarial Defense Layer (WLADL)

Inputs: $X = [x_1, x_2, ..., x_n]$: Tokenized input document
TH: WordNet (Fellbaum, 1998) Thesaurus
p_1: Synonym Altering Probability
p_2: Masking Probability
Output: $\hat{X}$: Altered input document

for $i \leftarrow 1$ **to** n **do**
  mask $\sim$ *Bernoulli*(p_2)
  if *mask* $= 0$ **then**
    synonym $\sim$ *Bernoulli*(p_1)
    if *synonym* $= 1$ **then**
      synonyms $\leftarrow$ TH.get[x_i]
      if *len(synonyms)* > 0 **then**
        index $\sim$ *Uniform*(1, len(synonyms))
        $\hat{x}_i \leftarrow$ synonyms[index]
      else
        $\hat{x}_i \leftarrow x_i$
    else
      $\hat{x}_i \leftarrow x_i$
  else
    $\hat{x}_i \leftarrow$ "" ▷ empty string
$\hat{X} \leftarrow [\hat{x}_1, \hat{x}_2, ..., \hat{x}_n]$
return $\hat{X}$

selected PWWS (Ren et al., 2019) and Genetic Algorithm (GA) (Alzantot et al., 2018), while to represent the ones that use language models, we chose BAE-R (Garg and Ramakrishnan, 2020). For detailed explanations about the attacks, we refer the readers to the original papers.

2.4 Experimental Pipeline

The candidate models were trained by applying either one of the proposed defense strategies, or trained cleanly, i.e. without any defense. Since finding an adversarial example is a computationally heavy operation, we followed the convention in the literature (Wang et al., 2021c,a,b) and attacked the first 200 samples of each test set. We calculated the classification metrics for the selected samples, both clean and attacked, with respect to every model and defense strategy. Overall, the pipeline we followed for the experiments can be seen in Figure 1.

2.5 Word-Level Adversarial Defense Layer (WLADL)

WLADL is a training-time algorithm that expects tokenized documents as input and outputs random documents generated by either masking or altering a token with its synonym using the WordNet (Fellbaum, 1998) thesaurus, also provided as input. The regularization is user definable by setting the synonym altering probability (p_1) and the masking probability (p_2). We observed that high values for p_1 and p_2 decrease clean performances. Therefore, we used and recommend $p_1 = 0.25$ and $p_2 = 0.1$. In Section 3.6, we present a comparative study for the effect of changing p_1. The corresponding pseudocode can be reviewed in Algorithm 1.

Our code can be found in the following GitHub repository: *Link omitted for anonymity*.

3 Results

Initially, we trained the candidate models on the selected datasets and reported test accuracy, area-under-ROC curve, and weighted F1 score. By attacking the clean trained models with PWWS, GA and BAE-R, we demonstrated the vulnerability to adversarial attacks. Afterwards, we applied the baseline defense strategies and WLADL, retrained the models from scratch and reported test metrics again, to ensure that accuracies acquired in clean training are maintained. Finally, the defended models were attacked with the same approach to monitor and compare how the defense algorithms improve robustness. We also compare defense algorithms with clean models in terms of accuracies under attack and number of queries generated by adversaries.

3.1 Clean Results

As expected, fine-tuned BERT outperforms BiLSTM and CNN in terms of all metrics (except AUROC on Yahoo! Answers). Generally, BiLSTM is the second-best model, followed by CNN. The clean classification metrics are included in Table 3.

3.2 Attacking Clean Models

We present attack results on the undefended models on 200 test samples, reporting the model accuracy for unperturbed samples, attacked samples, and the average number of queries generated by the adversaries in Table 2. The latter is a performance indicator for the adversary (the lower, the better).

Table 2: Attack results on the undefended candidate models. Best scores for the defendant are highlighted.

Model	Dataset	Original Accuracy	PWWS-Accuracy	BAE-Accuracy	GA-Accuracy	PWWS-Query	BAE-Query	GA-Query
BiLSTM	IMDb	0.885	0.0	0.215	**0.23**	1394.645	410.355	**3675.825**
CNN		0.8	0.0	0.07	0.055	1208.145	388.270	6330.29
BERT		**0.93**	**0.075**	**0.315**	–	**1514.015**	**417.75**	–
BiLSTM	AG News	0.895	0.165	0.745	**0.635**	318.26	147.755	**8834.555**
CNN		0.885	0.255	0.675	0.62	310.34	**208.645**	3472.805
BERT		**0.925**	**0.355**	**0.785**	–	**363.405**	128.765	–
BiLSTM	Yahoo! Answers	0.655	0.07	0.365	**0.27**	432.115	225.045	**17567.63**
CNN		0.575	0.055	0.235	0.155	363.655	227.755	15240.99
BERT		**0.665**	**0.235**	**0.485**	–	**538.26**	**306.345**	–

Table 3: Clean test metrics for candidate models

Model	Dataset	Accuracy	AU-ROC	F1
BiLSTM	IMDb	0.8033	0.8885	0.8018
	AG News	0.9022	0.9732	0.9023
	Yahoo! Answers	0.7092	**0.9328**	0.7027
CNN	IMDb	0.8004	0.8843	0.8004
	AG News	0.8896	0.9717	0.8895
	Yahoo! Answers	0.6311	0.8986	0.6224
BERT	IMDb	**0.9166**	**0.9711**	**0.9166**
	AG News	**0.9172**	**0.9803**	**0.9170**
	Yahoo! Answers	**0.7474**	0.9274	**0.7424**

We have also observed that it is easier to find perturbation for longer documents. This explains the better adversary performance on the IMDb dataset, which comprises longer documents on average.

While BERT is the most robust model overall, it also suffers from adversarial attacks, especially on the IMDb dataset, as its test accuracy drops from 0.93 to 0.075 (PWWS) and to 0.315 (BAE-R). The less sophisticated models, BiLSTM and CNN, are even more vulnerable. We also observe that with its low query number and better adversarial performance, PWWS is the strongest attack. BAE-R searches for fewer adversaries per sample while still harming the models, leaving GA as the least powerful attack.

3.3 Clean Results of Defended Models

To ensure performance maintenance, we measured the test metrics after training the candidate models using the defense strategies. The complete results can be found in Table 4. In general, performance is maintained, while WLADL and VAT defense show minor performance fluctuations. However, LSTM and CNN models experience performance drops on SEM training. This confirms that reducing the vocabulary in the embedding space decreases the clean performance.

3.4 Attacking Defended Models

The defended models were attacked using PWWS, BAE-R and GA. Additionally, to assess defense strategies' performances, we report average accuracy alterations to the models trained without any defense strategy in Table 5.

Since VAT samples were generated through attacking a clean trained BiLSTM using PWWS, we observed that the best defense strategy for BiLSTM was also VAT. We assume the hypothesized transferability issue of VAT to be true, as WLADL outperforms VAT on CNN and BERT models. SEM's inferior robustness is expected, due to the limitations imposed on the vocabulary. The embedding matrix used for clean training is GloVe with dimension 50 (400k tokens in the vocabulary), but for SEM, the vocabulary was restricted to 50k tokens per dataset. Using only the most frequent to-

Table 4: Clean Test Metrics for Candidate Models when trained with selected defense strategies

Model/Defense	Dataset	Accuracy	AU-ROC	F1
BiLSTM – WLADL	IMDb	0.769	0.860	0.763
	AG News	0.902	0.974	0.901
	Yahoo! Answers	0.710	**0.933**	0.705
BiLSTM – VAT	IMDb	**0.811**	**0.893**	**0.809**
	AG News	0.901	**0.975**	0.900
	Yahoo! Answers	**0.715**	0.931	**0.709**
BiLSTM – SEM	IMDb	0.781	0.856	0.781
	AG News	**0.903**	0.974	**0.903**
	Yahoo! Answers	0.700	0.928	0.695
CNN – WLADL	IMDb	0.789	0.870	0.789
	AG News	0.881	0.970	0.880
	Yahoo! Answers	0.624	0.899	0.611
CNN – VAT	IMDb	**0.814**	**0.895**	**0.813**
	AG News	**0.888**	**0.971**	**0.887**
	Yahoo! Answers	**0.633**	**0.902**	**0.625**
CNN – SEM	IMDb	0.772	0.857	0.772
	AG News	0.883	0.969	0.882
	Yahoo! Answers	0.622	0.894	0.616
BERT – WLADL	IMDb	0.882	0.963	0.880
	AG News	**0.915**	0.975	**0.914**
	Yahoo! Answers	0.743	0.927	0.735
BERT – VAT	IMDb	**0.921**	**0.974**	**0.921**
	AG News	**0.915**	**0.977**	**0.914**
	Yahoo! Answers	**0.746**	**0.928**	**0.740**

Table 5: Adversarial Accuracies of different defense strategies for models and datasets, averaged over attacks and compared against clean training.

Model	Dataset	$\Delta^{\text{Acc.}}_{\text{WLADL}}$	$\Delta^{\text{Acc.}}_{\text{VAT}}$	$\Delta^{\text{Acc.}}_{\text{SEM}}$
BiLSTM	IMDb	0.027	**0.053**	-0.028
	AG News	0.032	**0.101**	-0.020
	Yahoo! Answers	**0.055**	0.033	-0.038
CNN	IMDb	**0.119**	0.018	0.020
	AG News	0.014	**0.038**	-0.141
	Yahoo! Answers	**0.030**	0.020	-0.070
BERT	IMDb	**0.338**	0.047	–
	AG News	**0.063**	0.057	–
	Yahoo! Answers	**-0.006**	-0.052	–

kens leaves SEM more vulnerable to perturbations with less common tokens. WLADL showed to be most effective for CNN and BERT on the IMDb dataset, improving the average attacked accuracies by 0.12 and 0.34 respectively. Nonetheless, it is fair to claim that the defense strategies (both the selected baselines and WLADL) are weaker than the attacks.

3.5 Comparing Adversary Query Counts

To analyze the relationship between the defense mechanism and the difficulty to find adversarial examples, we reported query counts generated by adversaries for all possible combinations.

Table 6: Adversary Query Count changes with respect to different defense strategies, averaged over datasets & compared against clean training.

Model	Attack	$\Delta^{\text{Query}}_{\text{WLADL}}$	$\Delta^{\text{Query}}_{\text{VAT}}$	$\Delta^{\text{Query}}_{\text{SEM}}$
BiLSTM	PWWS	**62.52**	33.70	-100.46
	BAE	24.22	**29.20**	-11.68
CNN	PWWS	-1.26	**6.21**	-100.51
	BAE	-17.57	**-4.59**	-122.89
BERT	PWWS	**150.77**	19.04	–
	BAE	**49.20**	10.40	–

The averaged results in Table 6 are affirmative to the ones observed individually. An increase in query counts implies that searching further for perturbations is necessary to fool the classifiers, which is enforced by the defense. Again WLADL and VAT show similar statistics and perform mostly better than clean training, whereas attacking SEM proved less difficult regarding the change in query counts. BERT benefits the most from WLADL as adversaries have to search more on average.

3.6 Effect of Synonym Altering Probability

To better comprehend how altering the Synonym Altering Probability (p_1) effects defense performance, we trained the CNN with WLADL selecting $p_1 \in \{0.2, 0.4, 0.6, 0.8\}$ while controlling for other hyperparameters using the AG News dataset, and attacked them using PWWS. The results are presented in Table 7.

Table 7: Effect of changing WLADL Synonym Altering Probability on CNN, using AG News dataset, attacked by PWWS.

p_1	Acc.	PWWS-Acc.	Def. Suc.	Avg. Query
0.2	**0.849**	0.234	0.275	**262.68**
0.4	0.843	**0.236**	**0.279**	258.11
0.6	0.846	0.23	0.272	260.26
0.8	0.845	0.214	0.253	255.75

Diverse values for p_1 do not impact the unperturbed accuracy. However, when $p_1 > 0.4$, the perturbed accuracy drops, indicating that over-regularization also decreases the defense success.

4 Discussion

In this paper, we aimed to build a transferable and computationally efficient defense strategy against word-level black-box adversarial attacks for document classification. Existing strategies usually fail to exhibit those two qualities simultaneously. They focus on a certain class of models, augmenting the dataset and/or generating new embeddings, making these defenses non-transferable. The main strength of our defense strategy is applicability to any model/dataset without pretraining or computational burden, and ease of application as a side benefit. It is inherently synchronous to the training pipeline, avoiding any offline procedures like the existing methods.

Future work may include adapting WLADL's parameters dynamically during runtime, considering the length of the document and other variables, instead of being static inputs. The IMDb dataset, in particular, is characterized by longer documents than AG News and Yahoo! Answers. Thus, it could improve the poor performance against PWWS observed for this dataset with the CNN and BiLSTM models. Furthermore, we believe that a combination of VAT with WLADL where adversarial samples are not altered can be the most robust and the simplest choice to apply for document classification tasks.

References

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896, Brussels, Belgium. Association for Computational Linguistics.

Yong Cheng, Lu Jiang, and Wolfgang Macherey. 2019. Robust neural machine translation with doubly adversarial inputs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4324–4333, Florence, Italy. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. HotFlip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36, Melbourne, Australia. Association for Computational Linguistics.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. The MIT Press.

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)*, pages 50–56.

Siddhant Garg and Goutham Ramakrishnan. 2020. BAE: BERT-based adversarial examples for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6174–6181, Online. Association for Computational Linguistics.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR (Poster)*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Andrew L. 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, Portland, Oregon, USA. Association for Computational Linguistics.

John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. TextAttack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126, Online. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

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, Florence, Italy. Association for Computational Linguistics.

Suranjana Samanta and Sameep Mehta. 2017. Towards crafting text adversarial samples.

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, Online. Association for Computational Linguistics.

Boxin Wang, Shuohang Wang, Yu Cheng, Zhe Gan, Ruoxi Jia, Bo Li, and Jingjing Liu. 2021a. InfoBERT: Improving robustness of language models from an information theoretic perspective. In *International Conference on Learning Representations*.

Xiaosen Wang, Hao Jin, Yichen Yang, and Kun He. 2021b. Natural language adversarial defense through synonym encoding. *Conference on 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 the AAAI Conference on Artificial Intelligence*, 35(16):13997–14005.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1298–1308, Minneapolis, Minnesota. Association for Computational Linguistics.

A Algorithmic Pseudo-codes

Here, we give the pseudocode for the Synonym Encoding Method (Wang et al., 2021b) as we used it in our implementation.

A.1 Synonym Encoding Method

SEM modifies the word embeddings. It reduces an existing embedding matrix by mapping *similar* words to the most used one. Similarity is measured using the Euclidean distance in the embedding space. The pseudocode can be observed in Algorithm 2.

In our benchmarks, we followed the authors and used $\delta = 3.1$ and $k = 10$. The new embedding matrices were generated using the most frequent 50k tokens of each dataset.

Algorithm 2: Synonym Encoding Algorithm (Wang et al., 2021b)

Inputs: W: dictionary of words
n: size of W
δ: distance for synonyms
k: maximal number of synonyms for each word
Output: E: new embedding matrix

$E = \{w_1 : \text{NONE}, \ldots, w_n : \text{NONE}\}$
Sort the dictionary W by word frequency
for *each word* $w_i \in W$ **do**
  if $E[w_i] = NONE$ **then**
    if $\exists \hat{w}_i^j \in Syn(w_i, \delta, k), E[\hat{w}_i^j] \neq NONE$ **then**
      $\hat{w}_i^* =$ closest synonym to $w_i | \hat{w}_i^* \in Syn(w_i, \delta, k), E[\hat{w}_i^*] \neq$ NONE $E[w_i] = E[\hat{w}_i^*]$
    else
      $E[w_i] = w_i$
  for *each word* $\hat{w}_i^j \in Syn(w_i, \delta, k)$ **do**
    if $E[\hat{w}_i^j] = NONE$ **then**
      $E[\hat{w}_i^j] = E[w_i]$
return E

B Experimental Setup

We wrote our code in Python using the well-known Deep Learning framework PyTorch (Paszke et al., 2019). Attacking was done using the open-source library Text-Attack (Morris et al., 2020) which already provides recipes for the attacks.

B.1 Hyper-Parameters

The BiLSTM model we used consists of 64 hidden units with one bidirectional layer. The CNN was built from three blocks of 2D convolutions with kernel dimensions and filters [(2,50), (3,50), (4,50)], [3,5,7] respectively with 0.2 dropout. Both were trained using Adam (Kingma and Ba, 2015) for five epochs and default optimizer settings as given by PyTorch. For BERT, we fine-tuned the last two stacks of BERT-Base (see (Devlin et al., 2019)) with AdamW (decoupled weight decay version of Adam (Loshchilov and Hutter, 2019)) for three epochs. Here, the learning rate was set to $3 \cdot 10^{-5}$ and the biases were not corrected, while every other parameter was kept as default.

C Generating Adversarial Examples for VAT

For vanilla adversarial training, we augment the datasets using adversarial examples generated using a BiLSTM that was attacked by PWWS. For the IMDb and AG News datasets, we generated approximately 10% samples of the whole dataset, while for Yahoo! Answers as many as we could in 24 hours because of the computational burden. The amount of samples that were generated and how long it took, can be observed in Table 8.

Table 8: Augmented Adversarial Examples and their computational duration using PWWS and BiLSTM

Dataset	Duration	Samples Generated	Fraction of Training Set
IMDb	23:15h	2211	8.84 %
AG News	09:35h	12107	10.01 %
Yahoo! Answers	24:00h	13687	0.98 %

For examples of concrete adversarial samples that were generated, we refer the reader to Table 1.

D Attacking Defended Models

Following the complete pipeline in the main manuscript, we generated query and accuracy results for every possible defense, model, dataset, attack combination. The results can be seen in Table 9. The bolded ones are column-wise best results with respect to all defense strategies (the higher, the better).

Table 9: Attack Results on Defended Candidate Models with VAT, SEM and WLADL, bolded results imply the best defense performances

Defense Strategy	Model – Dataset	Original Accuracy	PWWS-Accuracy	BAE-Accuracy	GA-Accuracy	PWWS-Query	BAE-Query	GA-Query
VAT	BiLSTM – IMDb	0.905	0.0	**0.25**	**0.355**	1404.32	418.16	2999.635
	CNN – IMDb	**0.82**	0.0	0.085	0.095	234.015	383.465	**9282.585**
	BERT – IMDb	0.925	0.165	0.32	–	1582.775	440.5	–
	BiLSTM – AG News	**0.935**	**0.335**	**0.805**	0.71	**356.625**	**166.35**	3444.97
	CNN – AG News	**0.9**	**0.325**	**0.695**	0.645	**330.765**	**214.155**	3541.37
	BERT – AG News	**0.905**	0.425	**0.81**	–	360.535	143.02	–
	BiLSTM – Yahoo! Answers	**0.685**	0.115	0.38	**0.31**	485.185	286.24	1926.375
	CNN – Yahoo! Answers	**0.58**	0.075	0.225	0.205	**385.73**	**213.26**	1437.53
	BERT – Yahoo! Answers	**0.69**	0.175	0.44	–	529.515	300.57	–
SEM	BiLSTM – IMDb	0.805	0.0	0.18	0.18	1222.38	414.875	1986.455
	CNN – IMDb	0.705	0.005	0.055	0.125	1032.99	254.155	1702.43
	BiLSTM – AG News	0.92	0.16	0.745	0.58	313.49	168.93	3347.03
	CNN – AG News	0.85	0.195	0.325	0.61	273.915	134.895	3495.3
	BiLSTM – Yahoo! Answers	0.555	0.04	0.295	0.255	307.765	164.28	1753.5
	CNN – Yahoo! Answers	0.525	0.0	0.015	0.22	273.685	66.92	1508.355
WLADL	BiLSTM – IMDb	**0.925**	0.0	0.225	0.3	**1477.14**	**424.82**	**4374.865**
	CNN – IMDb	0.78	**0.01**	**0.145**	**0.325**	**1179.08**	**412.66**	2552.615
	BERT – IMDb	**0.96**	**0.47**	**0.595**	–	**1958.965**	**551.15**	–
	BiLSTM – AG News	0.9	0.245	0.75	0.645	331.155	139.41	**6426.63**
	CNN – AG News	0.875	0.29	0.675	0.625	325.745	163.46	**16304.345**
	BERT – AG News	0.905	**0.47**	0.795	–	**367.995**	**148.125**	–
	BiLSTM – Yahoo! Answers	0.655	**0.16**	**0.385**	**0.325**	524.28	**291.59**	**3046.7**
	CNN – Yahoo! Answers	0.555	**0.08**	**0.27**	0.185	373.535	195.83	**14867.37**
	BERT – Yahoo! Answers	0.665	**0.25**	**0.455**	–	**541.035**	**301.205**	–