Word-Level Adversarial Defense Layer for Robust Natural Language Classification

Noè Canevascini, Mert Erkul, Maximilian Herde, Yannick Schnider
{noec, merkul, herdem, yannicks}@ethz.ch

***Abstract*—Adversarial attacks, while being widely studied in Computer Vision (CV), have yet to be well defined and investigated in the Natural Language Processing (NLP) domain. However, recent results [1], [2], [3] indicate that NLP models for classification, sequence prediction and other tasks [4] are also prone to adversarial attacks. There have been some frameworks proposed to generate adversarial attacks and defense strategies, but they either modify the dataset by augmentation, create new embeddings or consist of complex algorithms, while being model and task dependent. In this study, we propose a modular, computationally efficient and architecture/dataset independent algorithm called the Word-Level Adversarial Defense Layer (WLADL) on CNN [5], Bidirectional LSTM and fine-tuned BERT [6] for document classification. For comparison, we also applied a vanilla adversarial training (VAT) [7] strategy (augmenting the dataset with adversarial examples), and the Synonym Encoding Method (SEM) [8] which generates new word embeddings. We evaluate the defense strategies through their clean test results, alterations in the accuracies and query counts compared to non-defended models. Our experiments demonstrate that WLADL shows competitive performance in defense to VAT.**

I. Introduction

Recently the robustness of Deep Learning (DL) models is being questioned due to adversarial perturbations which fool the models to make confident albeit false predictions or outputs [7]. These attacks are well studied in the CV domain [9] but have just recently gathered attention in NLP [10], [11]. In CV, generally [7], an adversarial example is a perturbation of an input which is imperceptible to the human eye, but fools the model. On the other hand, in textual inputs, as the concept of "imperceptibility" is not well defined [8], adversarial attacks are separated into four different classes: (i) character-level, (ii) word-level, (iii) sentence-level and (iv) multi-level attacks. Character-level attacks consist of intentional typos [10], [12], word-level attacks consist of low-frequency synonym replacement [13] for classification or antonym replacement for machine translation tasks [11], sentence-level attacks change word positions [14] and multi-level attacks add words to sentences for gradient disturbance [15]. In this study, we focused on examining defense strategies on word-level adversarial attacks for document classification tasks. One can observe some examples for word-level adversarial attacks in Table I.

Several problems arise while applying vanilla adversarial training (VAT) [7] and more complex schemes [8], [16]. To begin with, the VAT pipeline consists of training a clean model, attacking the clean model using a designated attack strategy, selecting successful adversarial examples and re-training the model by augmenting the training dataset with the adversarial examples. Transferability can be identified as the most significant issue in VAT. It is necessary to select a candidate model and an attack to complete the pipeline; one can suggest that an attack might be successful for one model but useless 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 Synonym Encoding Method (SEM) [8] or the Fast Gradient Projection Method (FGPM) [2] also suffer from transferability and computational burden. The SEM strategy consists of building new static word embeddings with respect to synonym distance, making BERT-like models not be able to fully utilise contextualized embeddings [6]. FGPM consists of calculating the Jacobian matrix with respect to inputs and model weights, causing a memory burden for overly parametrized models[1].

We therefore introduce a simple and architecture independent, stochastic training-time strategy: the Word-level Adversarial Defense Layer (WLADL). For every input, we either exchange a selected word with its synonym (using a pre-built WordNet [17] thesaurus), or mask it prior to feeding the input to the model aiming to increase its robustness. The probability of alteration (whether for synonym or for masking) is user definable, such that the proposed strategy works like a simple dropout layer.

Our final design goals for WLADL were as follows:

- Formalise an attack-independent defense approach to avoid the transferability issue of VAT and other more complex defense strategies.
- Avoid retraining models twice, once for generating adversarial samples and once after augmenting the dataset, and avoid building a new embedding space.
- Decrease the computational time and memory burdens imposed by the existing word-level defense strategies.

II. Models and Methods

To assess the performance of our approach, we selected the following baseline adversarial defense strategies, candidate models and datasets from the literature.

A. Models

- As being one of the most popular document classification models in the literature, the **Bidirectional Long-Short**

[1]Despite selecting FGPM as one of the baseline defense strategies in the proposal, because of the OOM issues we observed for BERT during implementation, we decided to replace it with VAT explained in [7].

TABLE I
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%)

Term Memory (BiLSTM) is chosen for performance evaluation.

- Even though **Convolutional Neural Networks (CNN)** are widely used in CV or Signal Processing tasks, due to their success in document classification [5], they are selected frequently as baseline classifiers in the adversarial NLP literature.
- After their introduction [6], fine-tuned **BERT** models have dominated downstream-NLP tasks surpassing other models in competitions. We opted for this model to monitor its ability to withstand adversarial attacks, and also to demonstrate how WLADL can be applied to it (and its counterparts such as RoBERTa [18]) while other defense strategies are either non-applicable or computationally heavy.

B. Datasets

- **IMDb:** This dataset consists of 25000 training and test samples of movie reviews, with positive or negative labels indicating the sentiment [19].
- **AG News:** This collection of four news classes, namely world, business, sports and science/technology, is comprised of balanced 120000 training and 7600 test examples [20].
- **Yahoo! Answers:** Having the most samples of the chosen datasets (1400000 training and 60000 testing samples, balanced with respect to classes), Yahoo! Answers consists of questions and corresponding answers, grouped based on topics into ten classes [20].

C. Baseline Defense Strategies

As a baseline defense algorithm, we selected VAT. We generated examples using a BiLSTM model trained on each of the clean datasets. Then we attacked the model using Probability Weighted Word Saliency [4] (PWWS), generating approximately 10% adversarial samples [7] for the IMDB and AG News training sets, and as many examples as possible in 24 hours for Yahoo! Answers. The detailed results for the number of samples generated and computational durations can be seen in Table II.

Another baseline we selected is the Synonym Encoding Method (SEM) [8], basically a modification of a static word embedding. It reduces an existing embedding matrix by mapping "similar" words to the most used one. We and the authors of the paper use eucledian distance of the word embeddings as indication of 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 we selected $\delta = 3.1$ and $k = 10$. We generated the new embedding matrices for every dataset using the most frequent 50000 tokens. The SEM algorithm [8] is given in Appendix A.

TABLE II
AUGMENTED ADVERSARIAL EXAMPLES AND THEIR COMPUTATIONAL DURATION USING PWWS AND BILSTM

Dataset	Duration	Samples Generated
IMDb	23:15:00	2211
AG News	09:35:00	12107
Yahoo! Answers	24:00:00	13687

D. Word-Level Adversarial Defense Layer (WLADL)

WLADL is a training-time algorithm that expects the tokenized document as input and outputs the random document generated by either masking or altering a token with its synonym using the WordNet [17] thesaurus, also provided as input. The regularization is user definable by setting the synonym altering probability (p_1) and the masking probability (p_2). The pseudo-code for WLADL can be seen in Algorithm 1. We observed that high values for p_1 and p_2 decrease clean performances. Therefore we used and recommend $p_1 = 0.25$ and $p_2 = 0.1$.

E. Attacks, Experimental Pipeline and Setup

1) Attacks: In literature, adversarial attacks are separated into two: white-box and black-box attacks. Black-box attacks do not make use of information regarding model parameters, whereas white-box attacks can also utilise them to perturb samples [1]. In our case 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 $C(X_i^{adv}) \neq y_i$ [1], [21]. Adversaries can employ external sources such as language-models, a WordNet [17] thesaurus and embeddings such as GloVe [22] to execute the attacks, but they have to respect certain constraints. Examples of such constraints are: the maximal number of

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

Inputs: $X = [x_1, x_2, ..., x_n]$: Tokenized input document
TH: WordNet [17] 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 $\hat{X}$

queries, maximal number of perturbations in a document and grammatical correctness [4], [1], [21]. These constraints were applied to ensure "minimal" perturbations. We selected three black-box attacks to assess model robustness and defense strategies' performances:

- **Probability Weighted Word Saliency (PWWS) [4]** orders words with respect to their output scores by replacing them with unknown tokens. After getting the word rankings (which the authors define as saliency [4]), the input text is perturbed through the saliency order, and it is greedily iterated to find the minimal adversarial perturbation (in terms of number of perturbed tokens) that affects the classifier score the most.
- **BAE-R [1]** replaces tokens in the document, by masking them first and then selecting the prediction of a BERT-Masked-Language-Model that alters the output score the most, while ensuring grammatical correctness.
- **Genetic Algorithm (GA) [21]** utilizes GloVe [22] embeddings to find the N-nearest neighbors of a word in the document and additionally uses Googles 1 billion words language model to eliminate out-of-context nearest neighbors of the candidates. Out of the remaining K samples, the word is replaced with the candidate that changes the model score the most through mutations and crossovers [21].

2) Experimental Pipeline: The candidate models were trained by applying either one of the proposed defense strategies, or trained cleanly without any defense. Using one of the selected attacks, the test dataset was attacked[2]. Since finding an adversarial example is a computationally heavy operation, we followed the convention in the literature [2], [16], [8] and attacked the first 200 samples of each test set. We calculated the accuracies for the selected samples, both clean and attacked, with respect to every model and defense strategy. Overall, the pipeline we followed for experiments can be seen in Figure 1.

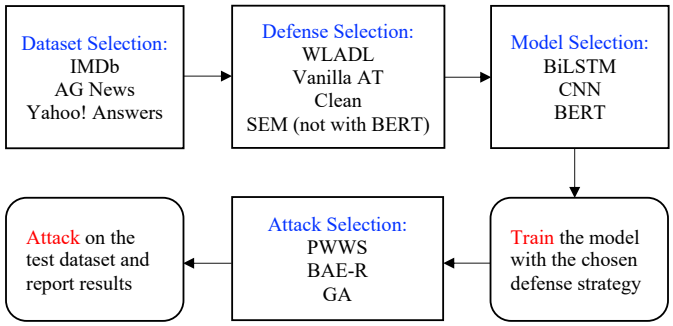

Fig. 1. Experimental Pipeline

3) Experimental Setup: We wrote our implementation in Python using the well-known DL framework PyTorch. Attacking was done using the open-source library TextAttack [23] which already provides recipes for the attacks. Our code can be found in the corresponding GitHub repository: https://github.com/maximilianherde/Word-level-adversarial-defense.

For work splitting reasons, both Google Colab and ETH Zurich's Euler cluster were used to train and attack the models.

III. Results

The results we obtained in this study can be split into five different subsections. Initially, we trained the candidate models on all selected datasets and obtained the test results for accuracy, area-under-ROC curve, and weighted F1 scores. Afterwards, we attacked the clean trained models with PWWS, GA and BAE-R to show that the candidate models are vulnerable to adversarial attacks. We then applied the baseline defense strategies and WLADL, trained the models from scratch and tested them once again, to ensure that accuracies acquired in clean training are maintained. Finally, we attacked the defended models with the same approach to monitor and compare how the defense algorithms improve robustness. We also compare defense algorithms with each other in terms of average accuracies under attack and average queries adversaries generate, taking the clean results as baselines. For the sake of completeness, all of the attack results are presented in Table VIII of Appendix B.

[2]We didn't attack BERT models using GA because of the heavy computational complexity, where attacking 200 samples takes more than three days of GPU time.

A. Clean Results

To obtain clean results, we trained BiLSTM and CNN models for 5 epochs, and fine-tuned the BERT classifier for 3 epochs. The hyperparameters used are as follows: for BiLSTM; 64 hidden units with 1 bidirectional layer, for CNN; three blocks of 2D convolutions of kernel dimensions and filters [(2,50), (3,50), (4,50)], [3,5,7] respectively with 0.2 dropout, and finally, we refer to [6] for fine-tuning the last two stacks of BERT-Base as a document classifier. The results for clean test metrics can be seen in Table III.

TABLE III
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**

As expected, fine-tuned BERT outperforms BiLSTM and CNN in terms of all metrics (except AU-ROC on Yahoo! Answers). Generally, BiLSTM is the second best model followed by CNN as the third on all datasets.

B. Attacking Clean Models

We present attack results on the undefended models on 200 samples for each test set and report the accuracy of the models for unperturbed samples, the accuracy attained when the sample is attacked, and the average number of queries generated by the adversaries. The average number of queries indicates the performance of the adversary in combination with the accuracy under attack. For both numbers it holds that lower is better for the adversary. We have also observed that the longer the input documents are, the easier it is to find a perturbation. This explains the better adversary performance for all attacks on the IMDB dataset where documents are longer compared to AG News and Yahoo! Answers, see Table IV.

Overall, it is possible to state that while BERT is the most robust model out of the candidate models, it also suffers from adversarial attacks, especially on the IMDb dataset, as its accuracy drops from 0.93 to 0.075 (PWWS) and to 0.315 (BAE-R) on the samples investigated. The simpler models, BiLSTM and CNN, suffer from attacks even more heavily, as accuracies that are more than 0.8 drop to 0, demonstrating their vulnerability. We also observe that with its low query number and better adversarial performance, PWWS is the strongest attack. BAE-R searches for less adversaries per sample while still harming the models, and GA is the weakest attack in terms of the extreme number of "potential adversaries" (queries) it generates and still not being as harmful compared to PWWS and BAE-R.

C. Defense Strategies & Their Clean Test Results

In order to ensure that the selected defense strategies did not perturb the clean results, we decided to report the test metrics using the same procedure conducted in Table III, after training the candidate models using the defense strategies (VAT, WLADL and SEM). The complete results (Table VII) can be found in Appendix B. In general, the defended models show similar results on the test set, where WLADL and VAT defense show minor performance fluctuations (VAT outperforms WLADL, as VAT training sets are larger), however, LSTM and CNN models experience performance drops on SEM training. This is expected and indicates that reducing the vocabulary in the embedding space decreases the clean performance.

D. Attacking Models with Defense Mechanism

The defended models were attacked using PWWS, BAE-R and GA. Table VIII in Appendix B contains all individual attack results. Additionally, to assess defense strategies' performances, we compared the changes in accuracy between the models trained without any defense strategy. The results for average accuracy alterations can be seen in Table V.

TABLE V
ADVERSARIAL ACCURACIES OF DIFFERENT DEFENSE STRATEGIES FOR MODELS AND DATASETS, AVERAGED OVER ATTACKS & 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	-

We observed that, as the VAT samples were generated through attacking a clean trained BiLSTM using PWWS, for BiLSTM, the best defense strategy on average was VAT. We may assume the hypothesised transferability issue of VAT to be true, as WLADL outperformed VAT on CNN and BERT models. We believe that the reason why SEM defense strategy is less robust than the other defense strategies and clean training itself, is because of the limitations imposed on the vocabulary. The embedding matrix used for clean training is the standard GloVe with 50 dimensions (400000 tokens in the vocabulary), but we limited the vocabulary to the 50000 most common tokens per dataset, for SEM. The reason is that computing the distance between pairs of tokens scales quadratically ($\mathcal{O}(|V|^2)$ with $|V|$ as size of the vocabulary) and we wanted to contain the runtime and memory. Using only the most frequent tokens leaves the models trained with 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

TABLE IV
ATTACK RESULTS ON THE UNDEFENDED CANDIDATE MODELS

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**	-

by 0.12 and 0.34 respectively. However, it is possible to claim that the defense strategies (both the selected baselines and WLADL) are insufficient. The decrease in accuracy because of PWWS (see Table VIII in Appendix B) are still detrimental to the models.

E. Comparing Adversary Query Counts

To address the question of how defense mechanisms make the adversaries' task of finding minimal perturbations more difficult, we also decided to compare the change in query counts generated by adversaries. The baseline query values are the query counts presented in Table IV, where averages were taken with respect to all datasets.

TABLE VI
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	-

Overall, it can be said that the results are affirmative to the ones observed in Table V. A positive change in query counts implies that the adversaries needed to generate non-minimal perturbations to fool the classifiers, caused by the presence of defense. Once again WLADL and VAT show similar performance and are mostly better than clean training, whereas attacking SEM was easier for adversaries in terms of the change in query counts. BERT is the model that benefits the most from WLADL, adversaries have to search further for perturbations over all datasets on average.

IV. DISCUSSION

In this study, we aimed to build a transferable, computationally efficient and model independent defense strategy against word-level black-box adversarial attacks for document classification. Existing strategies present two main challenges: transferability and computational cost. They consider a certain class of models to generate adversarial samples for augmentation and/or for generating new word embeddings, making these defenses non transferable to other models. The main strength of our defense strategy compared to others is applicability to any model/dataset without pretraining or computational burden, while not suffering from transferability issues as it is universal by design.

A feature to be added could be dynamically adapting synonym altering and masking probabilities during runtime depending on the length of the documents instead of being user-defined inputs. A reason supporting this suggestion is the poor performance against PWWS observed for the IMDb dataset with CNN and BiLSTM models. The IMDb dataset, in particular, is characterized by longer documents than AG News and Yahoo! Answers. Furthermore, we believe that a combination of VAT with WLADL where adversarial samples are not altered can be the most robust choice for document classification tasks and boost performance even further.

V. SUMMARY

We introduce problems in the existing adversarial defense strategies in NLP and propose a method, WLADL, to resolve some of them. We establish the candidate models, datasets, baseline defense algorithms and attacks which are applied through a pipeline. In the results section, we first analyse clean results to ensure that models are successful when not attacked, then show that they are vulnerable through systematically attacking them. Afterwards, to prove that the applied defense strategies do not change clean test results, we retrain the candidate models with defense, and compare test metrics of defense-based and clean training. In the end, we attack the defended models, compare the resulting accuracy and adversarial query counts with those of the non-defended attacks, ensuring a comparison between defense strategies. We show that while being transferable and computationally easy to implement, WLADL outperforms SEM and is additionally more universal. It also has better performance for attacks on CNN and BERT than VAT. In conclusion, we express the weaknesses of both baseline defense strategies and WLADL, and as future work recommend an input-length aware WLADL mixed with VAT as a possible defense strategy against black-box adversarial attacks for document classification.

REFERENCES

[1] S. Garg and G. Ramakrishnan, "Bae: Bert-based adversarial examples for text classification," 2020.

[2] X. Wang, Y. Yang, Y. Deng, and K. He, "Adversarial training with fast gradient projection method against synonym substitution based text attacks," 2020.

[3] C. Guo, M. Rana, M. Cisse, and L. van der Maaten, "Countering adversarial images using input transformations," 2018.

[4] S. Ren, Y. Deng, K. He, and W. Che, "Generating natural language adversarial examples through probability weighted word saliency," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 1085–1097. [Online]. Available: https://aclanthology.org/P19-1103

[5] Y. Kim, "Convolutional neural networks for sentence classification," 2014.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.

[7] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2015.

[8] X. Wang, H. Jin, Y. Yang, and K. He, "Natural language adversarial defense through synonym encoding," 2021.

[9] N. Akhtar, A. Mian, N. Kardan, and M. Shah, "Advances in adversarial attacks and defenses in computer vision: A survey," 2021.

[10] J. Ebrahimi, A. Rao, D. Lowd, and D. Dou, "Hotflip: White-box adversarial examples for text classification," 2018.

[11] Y. Cheng, L. Jiang, and W. Macherey, "Robust neural machine translation with doubly adversarial inputs," 2019.

[12] J. Gao, J. Lanchantin, M. L. Soffa, and Y. Qi, "Black-box generation of adversarial text sequences to evade deep learning classifiers," in *2018 IEEE Security and Privacy Workshops (SPW)*, 2018, pp. 50–56.

[13] S. Samanta and S. Mehta, "Towards crafting text adversarial samples," 2017.

[14] Y. Zhang, J. Baldridge, and L. He, "Paws: Paraphrase adversaries from word scrambling," 2019.

[15] L. Song, X. Yu, H.-T. Peng, and K. Narasimhan, "Universal adversarial attacks with natural triggers for text classification," 2021.

[16] B. Wang, S. Wang, Y. Cheng, Z. Gan, R. Jia, B. Li, and J. Liu, "Infobert: Improving robustness of language models from an information theoretic perspective," 2021.

[17] C. Fellbaum, *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

[18] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," 2019.

[19] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," Portland, Oregon, USA, pp. 142–150, June 2011. [Online]. Available: http://www.aclweb.org/anthology/P11-1015

[20] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," 2016.

[21] M. Alzantot, Y. Sharma, A. Elgohary, B.-J. Ho, M. Srivastava, and K.-W. Chang, "Generating natural language adversarial examples," 2018.

[22] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," Doha, Qatar, pp. 1532–1543, Oct. 2014. [Online]. Available: https://aclanthology.org/D14-1162

[23] J. Morris, E. Lifland, J. Y. Yoo, J. Grigsby, D. Jin, and Y. Qi, "Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp," pp. 119–126, 2020.

APPENDIX A
SYNONYM ENCODING ALGORITHM

Algorithm 2: Synonym Encoding Algorithm [8]

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 $w_i | \hat{w}_i^* \in Syn(w_i, \delta, k), E[\hat{w}_i^*] \neq \text{NONE}$
      $E[w_i] = E[\hat{w}_i^*]$
    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

APPENDIX B
ADDITIONAL RESULTS

TABLE VII
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 VIII
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**	-