Exploring the Vulnerability of Natural Language Processing Models via Universal Adversarial Texts

Xinzhe Li[1], **Ming Liu**[1,2], **Xingjun Ma**[1] and **Longxiang Gao**[1]

[1]School of IT, Deakin University, Australia
[2]Zhongtukexin Co. Ltd. , Beijing, China
{lixinzhe, m.liu,longxiang.gao}@deakin.edu.au
danxjma@gmail.com

Abstract

Universal adversarial texts (UATs) refer to short pieces of text units that can largely affect the predictions of Natural Language Processing (NLP) models. Recent studies on universal adversarial attacks require the availability of validation/test data which may not always be available in practice. In this paper, we propose two types of Data-Free Adjusted Gradient (DFAG) attacks to show that it is possible to generate effective UATs with manually crafted examples. Based on the proposed DFAG attacks, we explore the vulnerability of commonly used NLP models from two perspectives: network architecture and pre-trained embedding. The empirical results on three text classification datasets show that: 1) CNN-based and LSTM models are more vulnerable to UATs than self-attention models; 2) the vulnerability/robustness difference between of CNN/LSTM models and self-attention models could be attributed to whether or not they rely on training data artifacts for predictions; and 3) the pre-trained embeddings could expose vulnerability to both UAT and transferred UTA attacks.

1 Introduction

Deep neural networks (DNNs) have enabled significant advancement in a range of natural language processing (NLP) applications such as sentiment analysis (Yang et al., 2019; Xu et al., 2019) and topic classification (Sun et al., 2019). Despite the superior performance, DNNs are known to be vulnerable to adversarial perturbations (Szegedy et al., 2014; Goodfellow et al., 2015; Ma et al., 2018; Li et al., 2019; Ma et al., 2021), i.e., small changes on the input could lead to entirely incorrect predictions (Croce and Hein, 2020; Jiang et al., 2020). It has raised practical security concerns for the deployment of DNNs in safety-critical scenarios (Eykholt et al., 2018; Duan et al., 2020). Adversarially perturbed inputs are known as adversarial examples and the process of generating adversarial examples is known as adversarial attack. It has become a common practice to examine the vulnerability of DNNs to adversarial examples and mitigate the vulnerability by involving adversarial examples during the training process as a type of augmented data (Nie et al., 2020; Madry et al., 2018; Wang et al., 2019a; Zhang et al., 2019; Wang et al., 2019b; Croce et al., 2020).

Most adversarial attack methods for NLP models (Alzantot et al., 2018; Ebrahimi et al., 2018b; Jin et al., 2020) are sample-wise methods that craft adversarial examples by manipulating each clean example. Different from sample-wise attacks, universal adversarial attack (Behjati et al., 2019) aims to generate Universal Adversarial Texts (UATs) or universal triggers (Wallace et al., 2019) for each class or the entire dataset to fool NLP models. However, existing methods (Wallace et al., 2019; Song et al., 2021; Behjati et al., 2019) all require the validation/test dataset of the task or some proxy datasets in a similar domain to craft UATs.

To more easily and efficiently generate UATs, we propose Data-Free Adjusted Gradient (DFAG) attacks. According to the evaluation, our proposed DFAG attacks achieve a comparable performance as the original linear approximation method (Wallace et al., 2019) on most of the NLP models. We find that UATs generated by our method highly overlap with those from the original linear approximation method (Wallace et al., 2019). This indicates that the vulnerability of UATs may be inherent in the models. To better understand the vulnerability, we take text classification as an example and dive into different neural network architectures. Empirical results show that CNN and LSTM models are notebly more vulnerable to UATs than self-attention models. We also reveal that the effectiveness of UATs generated for LSTM and CNN

models exposes certain training data artifacts, i.e., important words in the training data that are more closely correlated with the targeted class. In contrast, self-attention models are relatively more robust to UATs. This finding is consistent with previous study on model robustness to training data artifacts, so it is likely that self-attention models suffer less from training data artifacts.

Apart from the neural architectures, we also examine pre-trained embeddings, including static pre-trained word embeddings (Pennington et al., 2014; Mikolov et al., 2018) and contextualized ones from the pre-trained language model BERT (Devlin et al., 2018). These embeddings have been widely used in different NLP applications. Our experiments show that pre-trained word embeddings could deteriorate model robustness to UATs, and even self-attention models can become vulnerable with pre-trained embeddings. Upon further investigation, we find that UATs are often transferable among models that use the same pre-trained embeddings. This reveals one unique vulnerability of NLP models to UATs.

2 Generating Universal Adversarial Texts

Problem Formulation. Consider a text classifier f mapping from input $\mathbf{x}$ to label y. The goal of universal adversarial attack is to generate a small sequence of tokens $\mathbf{t} = (t_1, t_2, ..., t_k)$ (i.e., an UAT), which can be inserted into any clean example x to cause misclassification towards a targeted wrong label $\tilde{y}$. Previous work (Behjati et al., 2019; Wallace et al., 2019) showed the effectiveness of UAT when three words are inserted at the beginning of the input sequence. Here, we follow their settings and predetermine the adversarial target class $\tilde{y}$. The attack problem can be formally definedd as: for any clean example $\{(\mathbf{x}, y)|(\mathbf{x}, y) \in \mathcal{D} \text{ and } y \neq \tilde{y}\}$, we aim to make the classifier f predict the perturbed example $\mathbf{t} \oplus \mathbf{x}$ as the targeted label $\tilde{y}$, i.e., $f(\mathbf{t} \oplus \mathbf{x}) = \tilde{y}$. The problem can be solved by minimizing an adversarial loss $\mathcal{L}_{adv}(\mathbf{t} \oplus \mathbf{x}, \tilde{y})$, which is the cross-entropy loss defined with the targeted label.

$$\arg\min_{\mathbf{t}} \quad \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[\mathcal{L}_{adv}(\mathbf{t} \oplus \mathbf{x}, \tilde{y})] \tag{1}$$

2.1 Gradient-based Attack

A UAT is composed of discrete tokens for which we search from the vocabulary $\mathcal{V} = w_1, w_2, \ldots, w_{|V|}$ ($|V|$ is the size of vocabulary). Each word w_i in the vocabulary is represented by a dense vector called embedding e_i. In order to find the optimal UAT, Behjati et al. (2019) applied gradient descent for $\mathbf{t}$ in the embedding space and identified the word in the vocabulary by projecting the nearest embeddings of the word. More efficiently, Wallace et al. (2019) proposed a linear approximation approach to generate gradients to approximate the loss of substituting $\mathbf{t}$ with $\mathbf{t}_{\text{update}}$, i.e., $\mathcal{L}_{adv}(\mathbf{t}_{\text{update}} \oplus \mathbf{x}, \tilde{y})$. According to the first-order Taylor approximation, we measure the effectiveness of the substitution by the inner product of the gradient $\nabla_{e_t}\mathcal{L}_{adv}$ with the embedding of t_{update}.

$$\arg\min_{e_t} \quad e_{t_{\text{update}}}^{\mathrm{T}} \nabla_{e_t}\mathcal{L}_{adv} \tag{2}$$

The approximation scores for all the possible substitution words in the vocabulary can be efficiently calculated via matrix multiplication, where $\mathbf{E} \in \mathbb{R}^{|V|\times m}$ denotes the embedding matrix with vocabulary size $|V|$ and embedding size m. It only needs one forward and backward pass to compute the gradients for all the positions of UAT tokens. The equation is shown below where $\nabla_{e_t}\mathcal{L}_{adv}$ has the dimensions for positions of UAT tokens and embedding size m.

$$\mathbf{A} = \mathbf{E} \times \nabla_{e_t}\mathcal{L}_{adv} \tag{3}$$

Both approaches require batches of data to update the UAT t. However, we can still use the linear approximation approach as a baseline for our experiment due to its efficiency. This approach requires a batch of examples to calculate the gradient for each update of the UAT, as shown in Equation (4) where n examples are consumed.

$$\nabla_{e_t}\mathcal{L}_{adv} = \frac{1}{n}\sum_{i=1}^{n} \nabla_{e_t}\mathcal{L}_{adv}(\mathbf{t} \oplus x_i) \tag{4}$$

2.2 Data-free Adjusted Gradient Attack

The universal property of UATs indicates that they reflect the inherent vulnerability of well-trained NLP models. Moreover, Wallace et al. (2019) reveals that UATs are a form of training data artifacts for natural language inference models. We suspect the validity of this conclusion across all text classification tasks, which is shown in Section 3.4.

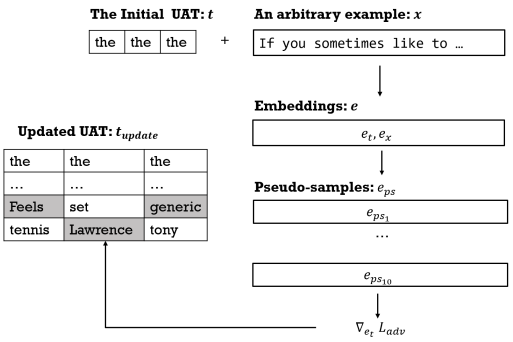

Figure 1: One iteration of the DFAG attack. The arbitrary example x is a positive movie review selected from the SST-2 test data, and the goal is to generate a UAT to make any non-negative (positive) reviews to be classified as negative ones. The UAT is generated by iterating the process: (1) concatenate the UAT t and the example x ; (2) generate dense text representation of $t \oplus x$, i.e., e_t, e_x; (3) generate pseudo-samples e_{ps} in the embedding space; (4) compute the gradient of adversarial loss w.r.t. e_t and finally find the updated UAT t_{updage} via the linear approximation method.

Therefore, using batches of data for universal attack might be redundant. Our proposed algorithm only requires an arbitrary example $\mathbf{x} = w_1, w_2, ..., w_l$ (l denotes the length of the text) which does not belong to the targeted class $\tilde{y}$ to generate effective UATs. In our experiment, we select the first valid sample from the test data. The attack could be *data-free* if the adversary chooses to manually craft the example. This is feasible because the only requirement for the example is that it does not belong to the targeted class. An interesting parallel work (Parekh et al., 2021) of data-free attack generates what they defined as "class impressions" for this purpose. We regard that the use of class impressions does lead to faster convergence but are not necessary, according to our experiments. Figure 1 demonstrates the process of updating the UAT in one iteration of our DFAG attack.

Unreliable gradients. The gradient for one single example might not be reliable since a DNN is usually not a smooth function. One most notable example is that an infinitesimal perturbation of the input could change its prediction. The issue also happens to the field of model interpretation, where they attribute input features for model prediction. Therefore, we generate pseudo-samples e_{ps} which are dense vectors $e_{ps_1}, .., e_{ps_K}$ in the embedding space and compute more reliable gradient by aggregating the gradients of the pseudo-samples.

Generating pseudo-samples. We pass the $\mathbf{t} \oplus \mathbf{x}$ into the embedding layer, which outputs the dense representation $\mathbf{e}$ in the embedding space. We then manipulate $\mathbf{e}$ to generate K pseudo-samples $\mathbf{e}_{ps}$ in the embedding space during each iteration. The gradients of the pseudo-samples are then aggregated to apply the linear approximation attack. We refer to this approach as the DFAG (Data-Free Adjusted Gradient) attack.

We employ the following two techniques to generate pseudo-samples, which have been proved to be effective in approximating gradients for model interpretation (Smilkov et al., 2017; Sundararajan et al., 2017).

- Smooth noise: the Gaussian noise η is generated with mean 0 and standard deviation σ. We denote this method as DFAG (Smooth) to accredit the *SmoothGrad* method (Smilkov et al., 2017).

$$\mathbf{e}_{ps} = \{\mathbf{e} + \eta_i \mid i \in [1..K]\} \quad \text{where} \quad \eta_i \sim \mathcal{N}(0, \sigma^2) \tag{5}$$

- Path method: we sample K pseudo-samples evenly along the straight path from the origin to the given sample. We denote this method as *DFAG (Integrated)* to accredit the *Integrated Gradient* method (Sundararajan et al., 2017).

$$\mathbf{e}_{ps} = \{\mathbf{e}_{ps_i} \mid i \in [1..K]\} \quad \text{where} \quad \mathbf{e}_{ps_i} = \frac{i}{K} \times \mathbf{e} \tag{6}$$

3 Attacking Text Classification Models

This section introduces model configurations and attack settings, and analyzes the experimental results. We also publish the source code for all the settings and experiments on Github [1] to reproduce the result.

3.1 Modeling Setup

Tasks and Datasets. Our experiments include Stanford Sentiment Treebank (SST-2) (Socher et al., 2013), Yelp (Zhang et al., 2015) datasets for sentiment classification task, and AG-News constructed by (Zhang et al., 2015) for topic classification task.

[1] https://github.com/xinzhel/attack_alta

Task	Architecture	Pre-trained	Attack Success Rate			ASR Ratio
			Baseline	DFAG (Smooth)	DFAG (Integrated)	
SST-2	LSTM	/	0.53	0.53	0.52	1
		GloVe	0.43	0.44	0.3	**1.02**
		FastText	0.85	0.85	0.81	1
		BERT	0.43	0.25	0	0.58
	CNN	/	1	0	1	1
		GloVe	1	1	1	1
		FastText	1	0	1	1
		BERT	0.25	0.2	0.1	0.8
	Self-attention	/	0.43	0.43	0.43	1
		GloVe	1	1	1	1
		FastText	1	1	1	1
		BERT	0.16	0.15	0	0.94
Yelp	LSTM	/	0.55	0.26	1	**1.82**
		GloVe	0.81	0.89	0.13	**1.1**
		FastText	0.58	0.4	0.24	0.69
		BERT	0.16	0.14	0.05	0.88
	CNN	/	1	0	0	*0*
		GloVe	0.91	0	0.37	*0.41*
		FastText	1	0	0.98	0.98
		BERT	0.24	0.14	0.02	0.58
	Self-attention	/	0.15	0.15	0	1
		GloVe	0.97	0.97	0.68	1
		FastText	0.98	0.98	0	1
		BERT	0.1	0.07	0.03	0.7
AG-News	LSTM	/	0.3	0.3	0.3	1
		GloVe	0.3	0.18	0	0.6
		FastText	0.2	0.2	0.2	1
		BERT	0	0	0	/
	CNN	/	1	1	1	1
		GloVe	0.88	0.91	0.41	**1.03**
		FastText	1	0.98	1	0.98
		BERT	0.04	0.06	0	**1.5**
	Self-Attention	/	0.02	0.01	0	0.5
		GloVe	0	0	0	/
		FastText	0.1	0.38	0.13	**3.8**
		BERT	0	0	0	/

Table 1: Attack success rates on different NLP models. One targeted class is selected to attack for each task: "negative" class for SST-2, Yelp, "Business" class for AG-News. The baseline attack refers to Wallace et al. (2019), while the DFAG (smooth) and DFAG (integrated) attacks use the smooth noise and path method. We use the ASR ratio of the DFAG attack to the baseline attack to measure the effectiveness of our DFAG attacks. **The numbers in bold** indicate that our DFAG attacks are more effective than the baseline. The ratios of less than 0.5 are marked in *the italics and underlining*, which indicate that our DFAG attacks are much less effective than the baseline.

Model architectures. We use three classical neural networks as the text classifiers.

- LSTM: Two-layer LSTM with 512 hidden dimensions. We take the final hidden state of the last time step for fully connected and softmax

layers to compute the probability distribution of all the classes.

- CNN (Zhang and Wallace, 2017): Four 1-dimensional convolution layers with filter sizes (2, 3, 4, 5) respectively. Each layer has six filters and is followed by the ReLU activation function and max-pooling layers. Therefore, the total output dimension is 24.

- Self-attention: One self-attention layer where we set 5 parallel attention heads (Vaswani et al., 2017) followed by a self-attentive pooling layer (McCann et al., 2017).

Pre-trained embeddings. We use static word embeddings GloVe (Pennington et al., 2014), FastText (Mikolov et al., 2018) and the contextualized embeddings from the last hidden layer of the pre-trained language model BERT (Devlin et al., 2018). The pre-trained embeddings can then be fed into the text classifiers. GloVe and FastText have different designs for obtaining word embeddings. GloVe embeddings are trained on a word co-occurrence matrix using a log-bilinear function where any pairs of word vectors are bilinearly mapped into the co-occurrence counts, while FastText embeddings are obtained by training a skip-gram model on word pairs from negative sampling.

All the pre-trained parameters are fixed without fine-tuning, as we aim to separate the vulnerability of the pre-trained embeddings from that of the model architectures and training. Specifically, we want to avoid propagating the information of the training data into pre-trained parameters, which would benefit the analyses of pre-trained embeddings and training data artifacts. In addition, when models use BERT embeddings with LSTM or CNN for classification, self-attention building blocks of BERT could interfere with our evaluation of architectures.

Training hyperparameters. We train all the models with the Adam optimizer, learning rate 5e-5, and batch size 64. The maximum number of training epochs is set to 5, and early stopping would occur when the validation accuracy has no improvement for one epoch.

3.2 Attack Setup

Attack hyperparameters. We select the first example from the attack data used by the baseline and then update the UATs in a maximum of 10 iterations with an early stop if there is no decrease of the loss $\mathcal{L}_{adv}$ for more than three iterations. We generate ten pseudo-samples during each iteration. The standard deviation of Gaussian noise is set as 0.01.

Constraints of substitution tokens. The vocabulary of BERT models has been built along with its pre-trained tasks, whereas we construct the word-level vocabulary from the training data for other models. Since sentiment words have strong indications for sentiment classification, sentiment words are filtered out following the practice in Wallace et al. (2019). In addition, our test examples are restricted to long sequences (>10 words) to preserve semantics to a large extent. BERT employs word-piece segmentation to process textual data into a sequence of sub-word units. However, when one or more sub-word are selected as the UAT tokens, the input may be re-segmented into a different sequence, such as the sub-word "##oot" which would be re-segmented into "#" "o" and "##ot". Our experiment shows that the word-level attack achieves similar performance, and tokens in the word unit cover 76.6% tokens in the BERT vocabulary. Therefore, we only consider substitution tokens in the word units to avoid the re-segmentation issue. The word-level substitutions also prevent that sub-words in UATs become unknown words during the UAT transfer attack.

Evaluation. We calculate Attack Success Rate (ASR) to measure the performance of the attack: the percentage of examples that are misclassified by the model as the targeted class among all the evaluation samples. We select evaluation examples that do not belong to the targeted class from the original test data.

3.3 Experimental Results

We first empirically verify the effectiveness of our attack on three neural network architectures, then evaluate the vulnerability of pre-trained embeddings via UAT transfer attacks.

Attack effectiveness. As shown in Table 1, our DFAG attacks with smooth gradients achieve competitive results on LSTM and self-attention models to the baseline. Moreover, the DFAG (Integrated) attack always performs better on CNN models, except the GloVe-CNN model on AG-News. Note that this finding does not involve BERT-based

models since BERT composes of multi-head self-attention layers.

To quantify how much effectiveness our DFAG attacks achieve relative to the baseline attack, we also report the ASR ratio of our DFAG attack to the baseline, i.e.,

$$\frac{\text{ASR of the DFAG}}{\text{ASR of the baseline}}$$

Here, we choose the better one between the two DFAG attacks. It shows that our DFAG attacks achieve more than 50% effectiveness of the baseline in most cases. An ASR ratio of more than 1 indicates that our DFAG (Smooth) attack even outperforms the baseline on several models. Note that our DFAG attacks are proposed to more easily and efficiently examine the vulnerability of NLP models to universal adversaries, rather than competing the ASR with existing attacks.

Failure cases on CNN models. Both DFAG attacks exhibit low success rates against CNN models on the Yelp dataset. By contrast, the baseline attack achieves nearly 100% success rates on all CNN models, where only the GloVe embeddings drop around 10% success rates on Yelp and AG-News datasets. This marks some failure cases of our DFAG attacks.

Comparing UATs generated by the baseline and our DFAG attacks. By comparing the UATs, we find that they actually generate many overlapped UAT tokens, especially for SST-2 models, as shown in Table 3. We suspect that the low overlap rates for AG_News and Yelp models are due to their large vocabulary sizes.

The vulnerability of pre-trained embeddings. As shown in Table 1, the use of pre-trained word embeddings sometimes makes the models more vulnerable, especially for self-attention models. This counter-intuitive result indicates the existence of embedding vulnerabilities in pre-trained embeddings. Our UAT transfer attacks also confirm the vulnerability of pre-trained embeddings. The result in Table 2 shows that UATs tend to achieve the best transferability on models with the same pre-trained embeddings. This phenomenon is also observed for BERT, although the success rate drops.

Measuring UAT transfer attacks. The absolute transfer ASR is not suitable to measure transferability because vulnerable models tend to have low ASRs. Therefore, in Table 2, we normalize the absolute transfer ASR by dividing by the original ASR of the victim model. The higher the normalized ASR the more transferable the UATs are to the target models (columns of Table 2). Take the first row as an example: the absolute transfer ASR of the BERT-LSTM model is only 0.06, while the vulnerable models always have higher ASRs. The normalized ASRs remove the effect of the varying vulnerabilities of the target models since it would amplify the absolute transfer ASR for the robust models, causing the value for BERT-LSTM from 0.06 to 0.44 (0.06 dividing by 0.14).

3.4 Training Data Artifacts in UATs

Training data artifacts are hypothesis words that are highly correlated with the labels. The artifacts have been explored by neural NLP models as the shallow shortcut and spurious correlations for the predictions (Gururangan et al., 2018; Branco et al., 2021). Wallace et al. (2019) argues that effective UATs for Natural Language Inference (NLI) models expose training data artifacts. Through our analyses, we further prove that training data artifacts should be attributed to the existence of UATs. Interestingly, we also find that the self-attention architecture provides certain robustness to such training data artifacts.

Measuring training data artifacts of UATs. We follow Gururangan et al. (2018); Wallace et al. (2019) and compute the point-wise mutual information (PMI) between each word w and the targeted class $\tilde{y}$ as:

$$\text{PMI}(w, \tilde{y}) = \log \frac{p(w, \tilde{y})}{p(w)p(\tilde{y})}$$

The denominator is the expected probability of the word w appearing in class $\tilde{y}$. The numerator is the observed probability. PMI measures how much more the word w occurs in the targeted class than we expect. We measure the training data artifacts of UAT words by their PMI ranks. We rank all the words according to their PMI scores in descending order. Then, the high-rank words show a high correlation with the targeted class, i.e., indicating training data artifacts. We also measure the frequency of each trigger word (i.e., the frequency in a particular class vs. the total frequency) because PMI would amplify words with low frequency.

Self-attention is robust to training data artifacts. The training data artifacts are highly reflected on UATs generated for CNN and LSTM

	Dataset	FastText			GloVe			BERT		
		LSTM	CNN	Self-Attention	LSTM	CNN	Self-Attention	LSTM	CNN	Self-Attention
FastText-LSTM	Yelp	1	0.8	0.42	0.2	0	0.05	**0.44**	0.08	0
	SST	1	1	0.93	**0.7**	**0.91**	**1**	0.02	0.04	0.12
GloVe-LSTM	Yelp	**0.31**	0.07	0	1	0.31	0.73	**0.43**	0.08	0
	SST	**0.96**	**1**	**0.82**	1	0.96	1	0.02	0.04	0.12
BERT-LSTM	Yelp	0.08	0.1	0.02	**0.37**	0.18	0.15	1	0.67	0.7
	SST	0	0.1	0.05	0	0.01	0.01	1	1.16	1.56

Table 2: The vulnerability of pre-trained embeddings is reflected by the UAT transfer attack. Rows: Each row represents the source models on which the UATs are generated. Columns: each column specifies a target model of the transfer attack. For example, the first row of the second column demonstrates the normalized ASR when we apply UATs generated on the FastText-LSTM model to the FastText-CNN model.

	Overlap Rates	Total Tokens	Overlap Tokens	vocabulary Size
SST-2	76%	21	16	17,356
AG-News	33%	6	2	114,068
Yelp	12%	8	1	746,663

Table 3: Overlap rates of the UATs generated by the baseline and our DFAG attacks.

models, while self-attention models generate UATs with low training data artifacts. The result is shown in Table 4. In order to verify the robustness of self-attention models to training data artifacts, the top 5 tokens with high training data artifacts are manually selected to evaluate the LSTM, CNN, and self-attention models. Only the self-attention model shows 0 attack success rates, as can be inferred from Table 5. The robustness of self-attention models may be attributed to their contextualized token representations: each token is represented by attending all the input tokens based on the attention scores. This type of architectures prevents the model from leveraging shallow shortcuts (class-wise triggers) for predictions.

4 Related Work

Universal adversarial perturbations. Behjati et al. (2019); Wallace et al. (2019); Song et al. (2021) generated the input-agnostic perturbations of text for NLP models. These works follow the initial work (Moosavi-Dezfooli et al., 2017) of finding Universal Adversarial Perturbations (UAPs) for images. Compared to the instance-specific adversarial perturbations (Liang et al., 2018; Ebrahimi et al., 2018b,a; Li et al., 2020), UAPs is a more severe security issue (Ribeiro et al., 2020). Behjati et al. (2019) employed projected gradient descent for devising UATs. Wallace et al. (2019) followed the linear approximation to generate adversarial text (Ebrahimi et al., 2018b) to generate UATs, which converges faster than Projected Gradient Descent (PGD). Song et al. (2021) generated natural UATs with less grammatical errors and more fluency via Adversarially Regularized Auto Encoder (ARAE). In this paper, we refer to the gradient approximation method. The original idea was proposed by Ebrahimi et al. (2018b) called Hotflip and then utilized by Wallace et al. (2019) to generate universal triggers.

***Gradient x Embedding* scores for model interpretation.** The first-order Taylor approach and *Gradient x Embedding* scores are also used to generate the saliency map in the field of model interpretation (Sundararajan et al., 2017; Li et al., 2016; Smilkov et al., 2017). However, they aim to attribute the softmax output of a neural network to input features while we identify the important words for substitutions in terms of adversarial loss $\mathcal{L}_{adv}$. Hence, the gradient is calculated for the output logits of the correct class rather than the adversarial loss, and also they use the embeddings of the original input instead of substitution words.

Adversarial transferability. Empirical study also mentioned the transferability of universal adversarial perturbations (UAPs) across models with distinguished architectures and pre-trained modules, such as image adversaries from VGG-19 to GoogleLeNet (Moosavi-Dezfooli et al., 2017) or ResNets to other networks (Wu et al., 2020), and adversarial texts from GloVe-based Reading Comprehensive models to ELMo-based models. In terms of explanations for adversarial transferability, Liang

Tokens	Models	Frequencies	PMI Ranks
"appears"	LSTM	11.0 / 11.0	3664
"Feels"	CNN	12.0 / 12.0	3665
"Lawrence"	CNN	11.0 / 12.0	4747
"pleasurable"	**Self-Attention**	**0.0 / 4.0**	**17181**
"unique"	LSTM	13.0 / 14.0	4990
"refreshingly"	CNN	10.0 / 10.0	4305
"mess"	**Self-Attention**	**1.0 / 30.0**	**15939**

(a) SST-2

Tokens	Models	Frequencies	PMI Ranks
"quickinfo"	LSTM	1813.0 / 1813.0	13250
"Qtr"	LSTM	62.0 / 63.0	15775
"hellip"	LSTM,CNN	80.0 / 80.0	13187
"Spitzer"	CNN	220.0 / 238.0	16114

(b) AG-News

As shown in Table 1, self-attention models are robust to UATs. Therefore, there are no effective UATs listed for self-attention models.

Tokens	Models	Frequencies	PMI Ranks
"giving"	LSTM	8184.0 / 12057.0	338822
"Horrible"	LSTM	4136.0 / 4158.0	311571
"inedible"	LSTM	2035.0 / 2108.0	311733
"Slowest"	CNN	117.0 / 117.0	311557
"BUYER"	CNN	97.0 / 97.0	309895
"disrespected"	CNN	216.0 / 217.0	311570
"restrain"	**Attention**	**8.0 / 41.0**	**735421**

(c) Yelp

Table 4: Training data artifacts of UAT tokens. Frequencies: In-class frequencies are displayed relatively to the total frequencies.

et al. (2020) proved its correlation with knowledge transferability, which relates to pre-trained knowledge. Also, adversarial transferability between imitated models and victim models (Wallace et al., 2020; He et al., 2021) also enhanced the relationship between pre-trained, transferable knowledge and adversarial transferability. These works motivate us to study the effect of pre-trained embeddings via the UAT transfer attack. Yuan et al. (2021) also studies the transferability of different architectures and pre-trained modules. Different from our study, they generate the sample-wise adversarial texts. Interestingly, they achieve an opposite conclusion that architecture types are more sensitive than pre-trained embeddings to transfer attacks.

5 Conclusion

In this work, we investigated the vulnerability of Natural Language Processing (NLP) models to Universal Adversarial Texts (UATs). We proposed two types of Data-Free Adjusted Gradient(DFAG) attacks which can generate effective UATs without real data. Our DFAG attacks lower the requirement of using UATs to understand the vulnerability of NLP models. With DFAG-generated UATs, we found that the robustness of self-attention to words with training data artifacts and revealed the unique (transferable) vulnerability of pre-trained embeddings. Our findings could help build robust NLP models against adversarial attacks. Future work could expose whether the pre-trained vulnerability

PMI Ranks	Models	ASR
1	LSTM	0.2
	CNN	0.1
	Self-Attention	**0**
2	LSTM	0.1
	CNN	0.1
	Self-Attention	**0**
3	LSTM	0.1
	CNN	0.1
	Self-Attention	**0**
4	LSTM	0.2
	CNN	0.4
	Self-Attention	**0**
5	LSTM	0.2
	CNN	0.5
	Self-Attention	**0**

Table 5: Evaluating the performance of SST models with the top-5 words out of the whole vocabulary according to their PMI ranks.

could make UATs transferable across different NLP tasks. Moreover, our result should also be verified on large-scale models. More detailed analyses of different filter sizes and attention heads are also interesting future works.

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.

Melika Behjati, Seyed-Mohsen Moosavi-Dezfooli, Mahdieh Soleymani Baghshah, and Pascal Frossard. 2019. Universal adversarial attacks on text classifiers. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7345–7349. IEEE.

Ruben Branco, António Branco, João António Rodrigues, and João Ricardo Silva. 2021. Shortcutted commonsense: Data spuriousness in deep learning of commonsense reasoning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1504–1521, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. 2020. Robustbench: a standardized adversarial robustness benchmark. *arXiv preprint arXiv:2010.09670*.

Francesco Croce and Matthias Hein. 2020. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pages 2206–2216. PMLR.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Ranjie Duan, Xingjun Ma, Yisen Wang, James Bailey, A Kai Qin, and Yun Yang. 2020. Adversarial camouflage: Hiding physical-world attacks with natural styles. In *IEEE/CVF conference on computer vision and pattern recognition*, pages 1000–1008.

Javid Ebrahimi, Daniel Lowd, and Dejing Dou. 2018a. On adversarial examples for character-level neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 653–663. Association for Computational Linguistics.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018b. Hotflip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, pages 31–36. Association for Computational Linguistics.

Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. 2018. Robust physical-world attacks on deep learning visual classification. In *IEEE conference on computer vision and pattern recognition*, pages 1625–1634.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R. Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 107–112. Association for Computational Linguistics.

Xuanli He, Lingjuan Lyu, Lichao Sun, and Qiongkai Xu. 2021. Model extraction and adversarial transferability, your BERT is vulnerable! In *Proceedings*

of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021, pages 2006–2012. Association for Computational Linguistics.

Linxi Jiang, Xingjun Ma, Zejia Weng, James Bailey, and Yu-Gang Jiang. 2020. Imbalanced gradients: A new cause of overestimated adversarial robustness. *arXiv preprint arXiv:2006.13726*.

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.

Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2019. Textbugger: Generating adversarial text against real-world applications. In *26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, California, USA, February 24-27, 2019*. The Internet Society.

Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2016. Visualizing and understanding neural models in NLP. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 681–691, San Diego, California. Association for Computational Linguistics.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. 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, Online. Association for Computational Linguistics.

Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. 2018. Deep text classification can be fooled. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4208–4215. International Joint Conferences on Artificial Intelligence Organization.

Kaizhao Liang, Jacky Y. Zhang, Oluwasanmi Koyejo, and Bo Li. 2020. Does adversarial transferability indicate knowledge transferability? *CoRR*, abs/2006.14512.

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. *arXiv preprint arXiv:1801.02613*.

Xingjun Ma, Yuhao Niu, Lin Gu, Yisen Wang, Yitian Zhao, James Bailey, and Feng Lu. 2021. Understanding adversarial attacks on deep learning based medical image analysis systems. *Pattern Recognition*, 110:107332.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6294–6305.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

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.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4885–4901. Association for Computational Linguistics.

Swapnil Parekh, Yaman Singla Kumar, Somesh Singh, Changyou Chen, Balaji Krishnamurthy, and Rajiv Ratn Shah. 2021. Minimal: Mining models for data free universal adversarial triggers. *arXiv preprint arXiv:2109.12406*.

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.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda B. Viégas, and Martin Wattenberg. 2017. Smoothgrad: removing noise by adding noise. *CoRR*, abs/1706.03825.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew 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, Seattle, Washington, USA. Association for Computational Linguistics.

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.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *China National Conference on Chinese Computational Linguistics*, pages 194–206. Springer.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing nlp. In *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, pages 2153–2162.

Eric Wallace, Mitchell Stern, and Dawn Song. 2020. Imitation attacks and defenses for black-box machine translation systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 5531–5546. Association for Computational Linguistics.

Yisen Wang, Xingjun Ma, James Bailey, Jinfeng Yi, Bowen Zhou, and Quanquan Gu. 2019a. On the convergence and robustness of adversarial training. In *International Conference on Machine Learning*, pages 6586–6595. PMLR.

Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. 2019b. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations*.

Dongxian Wu, Yisen Wang, Shu-Tao Xia, James Bailey, and Xingjun Ma. 2020. Skip connections matter: On the transferability of adversarial examples generated with resnets. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Hu Xu, Bing Liu, Lei Shu, and Philip Yu. 2019. BERT post-training for review reading comprehension and aspect-based sentiment analysis. 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 2324–2335, Minneapolis, Minnesota. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Liping Yuan, Xiaoqing Zheng, Yi Zhou, Cho-Jui Hsieh, and Kai-Wei Chang. 2021. On the transferability of adversarial attacks against neural text classifier. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1612–1625, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. 2019. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*, pages 7472–7482. PMLR.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 649–657.

Ye Zhang and Byron C. Wallace. 2017. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing, IJCNLP 2017, Taipei, Taiwan, November 27 - December 1, 2017 - Volume 1: Long Papers*, pages 253–263. Asian Federation of Natural Language Processing.