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

Motivation

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. We question that *whether it is possible to generate effective UATs with manually crafted examples*.

Data-Free Adjusted Gradient (DFAG) Attacks

How to calculate gradients? In order to compute more reliable gradient, we generate pseudo-samples which are dense in the embedding space and aggregate the gradients of the pseudo-samples

Why 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.

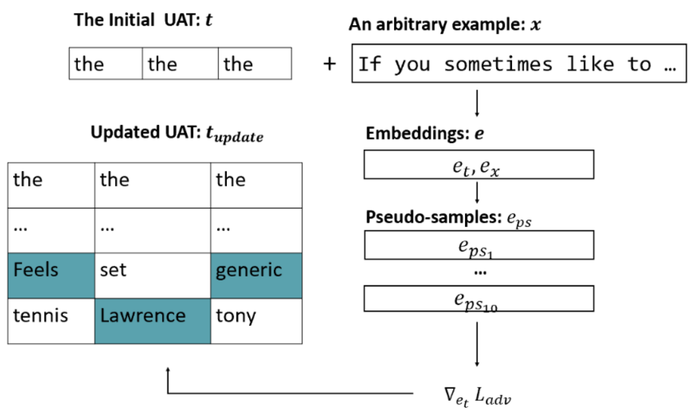

Figure 1: One iteration of the DFAG attack. The arbitrary example x is a positive movie review selected from the SST-2 test data, and the goal is to generate a UAT to make any non-negative (positive) reviews to be classified as negative ones. The UAT is generated by iterating the process: (1) concatenate 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.

Effectiveness of Data-Free Attacks: According to the evaluation, our proposed DFAG attacks achieve a comparable performance as the original linear approximation method on most of the NLP models. We find that UATs generated by our method highly overlap with those from the original linear approximation method . 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.

Vulnerability

We further explore where the vulnerabilities come from in terms of network architecture and pre-trained embedding on three text classification datasets.

We first evaluate the most commonly used architectures, showing that: 1) CNN-based and LSTM models are notably 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;

we also examine pre-trained embeddings, including static pre-trained word embeddings and contextualized ones from the pre-trained language model BERT . These embeddings have been widely used in different NLP applications. We find that pre-trained word embeddings could deteriorate model robustness to UATs, and even self-attention models can become vulnerable with pre-trained embeddings.

Transferability

Our experiments show that UATs are often transferable among models that use the same pre-trained embeddings. This reveals one unique vulnerability of NLP models to UATs

	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.

Training Data Artifacts

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.

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 r total frequencies.

DEAKIN UNIVERSITY