MINIMAL: Mining *Models* for Universal Adversarial Triggers

Yaman Kumar Singla[*1,3,5], **Swapnil Parekh**[*2], **Somesh Singh**[*3],
Balaji Krishnamurthy[1], **Rajiv Ratn Shah**[3], **Changyou Chen**[5]

[1]Adobe Media Data Science Research, [3]IIIT-Delhi,
[5]SUNY at Buffalo, [2]New York University

Abstract

It is well known that natural language models are vulnerable to adversarial attacks, which are mostly input-specific in nature. Recently, it has been shown that there also exist input-agnostic attacks in NLP models, special text sequences called universal adversarial triggers. However, existing methods to craft universal triggers are data intensive. They require large amounts of data samples to generate adversarial triggers, which are typically inaccessible by attackers. For instance, previous works take 3000 data samples per class for the SNLI dataset to generate adversarial triggers. In this paper, we present a novel data-free approach, *MINIMAL*, to mine input-agnostic adversarial triggers from models. Using the triggers produced with our data-free algorithm, we reduce the accuracy of Stanford Sentiment Treebank's positive class from 93.6% to 9.6%. Similarly, for the Stanford Natural Language Inference (SNLI), our single-word trigger reduces the accuracy of the entailment class from 90.95% to less than 0.6%. Despite being completely data-free, we get equivalent accuracy drops as data-dependent methods[1].

1 Introduction

In the past two decades, deep learning models have shown impressive performance over many natural language tasks, including sentiment analysis (Zhang, Wang, and Liu 2018), natural language inference (Parikh et al. 2016), automatic essay scoring (Kumar et al. 2019), question-answering (Xiong, Zhong, and Socher 2017), keyphrase extraction (Meng et al. 2017), *etc*. At the same time, it has also been shown that these models are highly vulnerable to adversarial perturbations (Behjati et al. 2019). The adversaries change the inputs to cause the models to make errors. Adversarial examples pose a significant challenge to the rising deployment of deep learning based systems.

Commonly, adversarial examples are found on a per-sample basis, *i.e.*, a separate optimization needs to be performed for each sample to generate an adversarially perturbed sample. Since the optimization needs to be performed for each sample, it is computationally expensive and requires deep learning expertise for generation and testing. Lately, several research studies have shown the existence of input-agnostic universal adversarial trigger (UATs) (Moosavi-Dezfooli et al. 2017; Wallace et al. 2019). These are a sequence of tokens, which, when added to any example, cause a targeted change in the prediction of a neural network. The existence of such word sequences poses a considerable security challenge since the word sequences can be easily distributed and can cause a model to predict incorrectly for all of its inputs. Moreover, unlike input-dependent adversarial examples, no model access is required at the run time for generating UATs. At the same time, the analysis of universal adversaries is interesting from the point of view of model, dataset analysis and interpretability (§5). They tell us about the global model behaviour and the general input-output patterns learnt by a model (Wallace et al. 2019).

Existing approaches to generate UATs assume that an attacker can obtain the training data on which a targeted model is trained (Wallace et al. 2019; Behjati et al. 2019). While generating an adversarial trigger, an attacker firstly *trains* a proxy model on the training data and then generates adversarial examples by using gradient information. Table 1 presents the data requirements during training for the current approaches. For instance, to find universal adversaries on the natural language inference task, one needs 9000 training examples. Also, the adversarial ability of a perturbation has been shown to depend on the amount of data available (Mopuri, Garg, and Radhakrishnan 2017; Mopuri, Ganeshan, and Babu 2018). However, in practice, an attacker rarely has access to the training data. Training data are usually private and hidden inside a company's data storage facility, while only the trained model is publicly accessible. For instance, Google Cloud Natural Language (GCNL) API only outputs the scores for the sentiment classes (Google 2021) while the data on which the GCNL model was trained is kept private. In this real-world setting, most of the adversarial attacks fail.

In this paper, we present a novel data-free approach for crafting universal adversarial triggers to address the above issues. Our method is to mine a trained *model* (but not data) for perturbations that can fool the target model without any knowledge about the data distribution (*e.g.*, type of data, length and vocabulary of samples, *etc*.). We only need access to the embedding layer and model outputs. Our method achieves this by solving first-order Tay-

[*]Equal Contribution

[1]The code and reproducibility steps are given in https://github.com/midas-research/data-free-uats

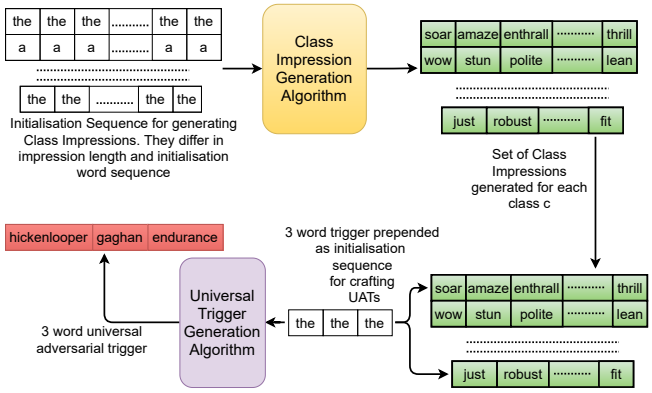

Figure 1: Two step process to generate universal adversarial triggers. First, we generate multiple class impressions for each class c. For this, we take multiple initialization sequences differing in starting word and length. After generating class impressions, we use them as our dataset for generating universal adversarial triggers.

lor approximation of two tasks: first, we generate "class-impressions" (§3.1), which are reconstructed text sentences from a model's memory representing the learned parameters for a certain data class; and second, we mine universal adversarial triggers over these generated class impressions (§3.2). Class-impression can be considered as the general representation of samples belonging to a particular class (Fig 5) and are used to emulate samples belonging to that class in our method. The concept of data leaving its impression on a trained model has also been observed in prior work in model inversion attacks in computer vision (Micaelli and Storkey 2019; Nayak et al. 2019). We build on that concept to mine universal adversarial triggers. We propose a combination of general model inversion attacks methodology with trigger generation to mine data-free adversarial triggers and show our results for several NLP models (Fredrikson, Jha, and Ristenpart 2015; Tramèr et al. 2016).

The major contributions of our work are summarized as:

- For the first time in the literature, we propose a novel data-free approach, MINIMAL (MINIng Models for AdversariaL triggers), to craft universal adversarial triggers for natural language processing models and achieve state-of-the-art success (adversarial) rates (§4). We show the efficacy of the triggers generated using our method on three well-known datasets and tasks, *viz.*, sentiment analysis (§4.1) on Stanford Sentiment Treebank (SST) (Socher et al. 2013), natural language inference (§4.2) on the SNLI dataset (Bowman et al. 2015), and paraphrase detection (§4.3) on the MRPC dataset (Dolan and Brockett 2005).
- We use both class impressions and universal adversarial triggers generated by our models to try to understand the models' global behaviour (§5). We observe that the words with the lowest entropy (*i.e.*, the most informative features) appear in the class impressions (Fig. 4). We find that these low entropy word-level features can also act as universal adversarial triggers (Table 12). The class-impression words are good representations of a class since they form distinct clusters in the manifold representations of each class.

2 Related Work

Universal Adversarial Attacks: Moosavi-Dezfooli et al. (2017) showed the existence of universal adversarial perturbations. They showed that a *single perturbation* could fool DNNs most of the times when added to all images. Since then, many universal adversarial attacks have been designed for vision systems (Khrulkov and Oseledets 2018; Li et al. 2019; Zhang et al. 2021). To the best of our knowledge, there are only three recent papers for NLP based universal adversarial attacks, and all of them require data for generating universal adversarial triggers (Wallace et al. 2019; Song et al. 2021; Behjati et al. 2019). In simultaneous works, (Wallace et al. 2019; Behjati et al. 2019) show universal adversarial triggers for NLP. Song et al. (2021) extend it to generate natural (data-distribution like) triggers. We compare our work with (Wallace et al. 2019) since they show improved adversarial success rates over (Behjati et al. 2019). We leave mining natural triggers from models as a future study. Our results demonstrate comparable performance as (Wallace et al. 2019) but without using any data. Table 1 mentions the data requirement of (Wallace et al. 2019).

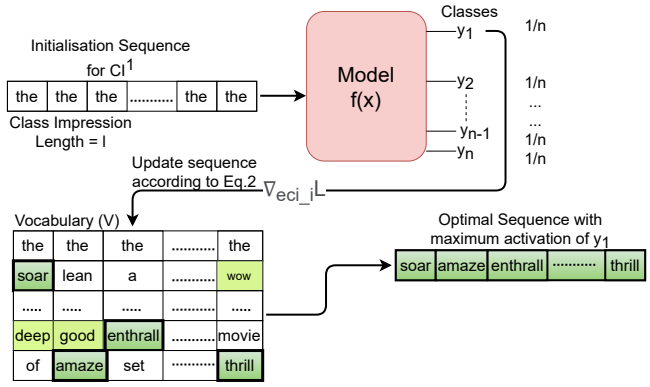

Figure 2: Class Impression Generation (CIG) Algorithm. We start with an initial sequence of *"the the ... the"* and continuously update it based on its gradient with respect to output probabilities (Eq. 2). The final sequence we get represents the class impression CI^c for the class c.

While there are many proposed classifications of adversarial attacks, from the point of view of our work, they can be seen in two ways: (a) data-based attacks; (b) data-free attacks. Data-based computer vision attacks depend on training and validation dataset to craft adversaries, while data-free attacks rely on other signals. There are some data-free approaches in computer vision, for example, by maximizing activations at each layer (Mopuri, Garg, and Radhakrishnan 2017; Mopuri, Ganeshan, and Babu 2018), class activations (Mopuri, Uppala, and Babu 2018), and pretrained models and proxy dataset (Huan et al. 2020). However, there has been no work in NLP systems for data-free attacks.

3 The Proposed Approach

In summary, our algorithm of crafting data-free universal adversarial triggers is divided into two steps, as shown in Fig 1.

First, we generate a set of class-impressions (§3.1) (Fig 2) for each class. These natural language examples represent the entire class of samples and are generated solely from the weights learnt by the model. Second, we use the set of class impressions generated in the first step to craft universal adversarial triggers corresponding to those impressions (§3.2) (Fig 3).

3.1 Class-Impressions Generation (CIG) Algorithm

To generate the class impression CI^c for a class c, we propose to maximize the confidence of the model $f(x)$ for an input text sequence $\mathbf{t}_c$. Formally, we maximize:

$$CI^c = \arg\max_{\mathbf{t}_c} \mathbb{E}_{\mathbf{t}_c \sim \mathcal{V}} \left[\mathcal{L}(c, f(\mathbf{t}_c)) \right], \tag{1}$$

where $\mathbf{t}_c$ is sampled from a vocabulary $\mathcal{V}$. The input $\mathbf{t}_c$ in NLP is not continuous, but is made up of discrete tokens. Therefore, we use the first-order Taylor approximation of Eq. 1 (Michel et al. 2019; Ebrahimi et al. 2018; Wallace et al. 2019). Formally, for every token $\mathbf{e}_{ci_i}$ in a class impression CI^c, we solve the following equation:

$$\mathbf{e}_{ci_i} = \arg\min_{\mathbf{e}'_i \in \mathcal{V}} [\mathbf{e}'_i - \mathbf{e}_{ci_i}]^\intercal \nabla_{\mathbf{e}_{ci_i}} \mathcal{L}, \tag{2}$$

where $\mathcal{V}$ represents the set of all words in the vocabulary, and $\nabla_{\mathbf{e}_{ci_i}} \mathcal{L}$ is the gradient of the task loss. We model the Eq. 2 as an iterative procedure by starting out with an initialisation value of $\mathbf{e}_{ci_i}$ as *'the'*. We then continually optimize it until convergence. For computing the optimal $\mathbf{e}'_i$, we take $|\mathcal{V}|$ d-dimensional dot products where d is the dimensionality of the token embedding. We use beam-search for finding the optimal sequence of tokens $\mathbf{e}'_i$ to get the minimum loss in Eq. 2. We score each beam using the loss on the batch in each iteration of the optimization schedule.

Finally, we convert the optimal $\mathbf{e}_{ci_i}$ back to their associated word tokens. Fig. 2 presents an overview of the process. It shows the case where we initialized $\mathbf{e}_{ci_i}$ with a sequence of *"the the .... the"* and then follow the optimization procedure for finding the optimal CI^c for the class c[2].

To generate class impressions for the models that use contextualized embeddings like BERT (Devlin et al. 2019), we perform the above optimization over character and sub-word level. We also replace the context-independent embeddings in Eq. 2 with contextual embeddings as obtained from BERT after passing the complete sentence to it.

We generate multiple class impressions for each class for all models by varying the number of tokens and the starting sequence. This gives us a number of class impressions for the next step where we generate triggers over these class impressions.

3.2 The Universal Trigger Generation (UTG) Algorithm

After generating class impressions in the previous step, we generate adversarial triggers as follows. From the last algorithm, we get a batch of class impressions CI^c for the class c. The task of crafting universal adversarial triggers is defined as minimizing the following loss function:

$$\arg\min_{\mathbf{t}_{adv}} \mathbb{E}_{\mathbf{t} \sim \mathcal{CI}^c} \left[\mathcal{L}(\tilde{c}, f(\mathbf{t}_{adv}; \mathbf{t})) \right], \tag{3}$$

where $\tilde{c}$ denotes target class (distinct from the class c), $f(\mathbf{t}_{adv}; \mathbf{t})$ denotes the evaluation of $f(x)$ on the input containing concatenation of adversarial trigger tokens at the start of the text t. The text t is sampled from the set of all class impressions CI^c. Again, we use the Taylor approximation of the above equation. Therefore, we get:

$$\mathbf{e}_{adv_i} = \arg\min_{\mathbf{e}'_i \in \mathcal{V}} [\mathbf{e}'_i - \mathbf{e}_{adv_i}]^\intercal \nabla_{\mathbf{e}_{adv_i}} \mathcal{L}, \tag{4}$$

where $\mathcal{V}$ represents the set of all words in the vocabulary, and $\nabla_{\mathbf{e}_{adv_i}} \mathcal{L}$ is the average gradient of the task loss over a batch. We model Eq. 4 as an iterative procedure where we initialize $\mathbf{e}_{adv_i}$ with an initialisation value of 'the'. For computing the optimal $\mathbf{e}'_i$, similar to the previous step, we take $|\mathcal{V}|$ d-dimensional dot products where d is the dimensionality of the token embedding. We use beam-search for finding the optimal sequence of tokens $\mathbf{e}'_i$ to get the minimum loss in Eq. 4. We score each beam using the loss on the batch in each iteration of the optimization schedule. Additionally, to generate impressions of varying difficulty, we randomly select the token from a N-sized beam of possible minimal candidates, instead of the least scoring candidate.

Finally, we convert the optimal $\mathbf{e}_{adv_i}$ back to their associated word tokens. Fig. 3 presents an overview of the process. Similar to Sec. 3.1, we initialize the iterative algorithm with a sequence ($\mathbf{e}_{adv}$) of *"the the .... the"*[3] and then follow the optimization procedure to find the optimal $\mathbf{e}_{adv}$. We handle contextual embeddings in a similar manner as in Sec. 3.1. Next, we show the application of the algorithms developed on several downstream tasks.

Dataset	Validation Size (Real samples)	Impressions Size (Generated samples)
SST	900	300
SNLI	9000	400
MRPC	800	300

Table 1: Number of Samples required to generate Universal Adversarial Triggers for each Dataset. In a data-based approach like (Wallace et al. 2019), validation set (column 2) is used to generate the UATs. The third column lists the number of queries we make to generate artificial samples. These artificial samples are then used to craft UATs. Note that no real samples are required for our method.

4 Experiments

We present our experimental setup and the effectiveness of the proposed method in terms of the success rates achieved by the crafted UATs. We test our method on several tasks including sentiment analysis, natural language inference, and paraphrase detection.

[2]We vary the initialization sequence and sequence length to generate multiple class impressions for the same class

[3]We vary the initialisation sequence and sequence length to generate multiple adversarial triggers

Class	Class Impression
Positive	energizes energizes captivated energizes enthrall eye-catching captivating aptitude artistry passion
Positive	captures soul-stirring captivates mesmerizing soar amaze excite amaze enthrall thrill captivating impress artistry accomplishments
Negative	spiritless ill-constructed ill-conceived ill-fitting aborted fearing bottom-rung woe-is-me uncharismatically pileup
Negative	laziest third-rate insignificance stultifyingly untalented hat-in-hand rot leanest blame direct-to-video wounds urinates

Table 2: Class Impressions for BiLSTM-Word2Vec Sentiment Analysis Model. Note that the words in the class impression examples highly correspond to the respective sentiment classes.

Type	Direction	Trigger	Acc. Before	Acc. After
Data-based	P → N	worthless endurance useless	93.6	9.6
Data-free	P → N	useless endurance useless	93.6	9.6
Data-based	N → P	kid-empowerment hickenlooper enjoyable	80.3	7.9
Data-free	N → P	compassionately hickenlooper gaghan	80.3	8.1

Table 3: The table reports the accuracy drop for the BiLSTM-Word2Vec sentiment analysis model after prepending 3-word adversarial triggers generated using MINIMAL and data-based methods.

Type	Direction	Trigger	Acc. Before	Acc. After
Data-free	P → N	useless endurance useless	86.2	32
Data-free	N → P	compassionately hickenlooper gaghan	86.9	35

Table 4: Accuracy drop for transfer attack with data-free UAT generated by our method. We prepend 3-word adversarial triggers to the SST BiLSTM-ELMo model.

4.1 Sentiment Analysis

We use the Stanford Sentiment Treebank (SST) dataset (Socher et al. 2013). Previous studies have extensively used this dataset for studying sentiment analysis (Devlin et al. 2019; Cambria et al. 2013). We use two models on this dataset: Bi-LSTM model (Graves and Schmidhuber 2005) with word2vec emebddings (Mikolov et al. 2018), Bi-LSTM model with ELMo embeddings (Peters et al. 2018). The same models have been used in previous work (Wallace et al. 2019) for generating data-dependent universal adversarial triggers. The models achieve an accuracy of 84.4% and 86.6% over the dataset, respectively. We compare our algorithm with (Wallace et al. 2019) since it is demonstrated to work better than other works (Behjati et al. 2019).

Class Impressions: First, we generate class impressions for the model. Table 2 presents 2 class impressions per class. As can be seen from the table, the words selected by the CIG algorithm highly correspond to the class sentiment. For instance, the algorithm selects positive words such as *energizes, enthrall* for the positive class, and negative words such as *spiritless, ill-conceived, laziest* for the negative class. We posit that the class impressions generated through our algorithm can be used to interpret what a model has learnt.

UAT: Next, using the class impressions generated for the models, we generate universal adversarial triggers with the UTG algorithm (Sec 3.2). In order to avoid selecting construct-relevant words, we remove such words[4] from our vocabulary for this task. Table 3 shows the results for the performance of adversarial triggers generated using our method and those by the data-based approach of (Wallace et al. 2019). Despite being completely independent of data, we achieve comparable accuracy drops as (Wallace et al. 2019). We are able to reduce the sentiment prediction accuracy by more than 70% for both the classes.

Transfer of Mined UATs: We check whether the triggers mined from one model also work on other models. For this, we test the triggers mined from BiLSTM-Word2Vec model on the BiLSTM-ELMo model. Table 4 notes the results for the same. The triggers reduce the accuracy for both the classes by more than 50%. This is significant since they are completely mined from the model without any information of the underlying distribution. We also compare the attack success rate as a function of trigger length (Fig. 6).

4.2 Natural Language Inference

For natural language inference, we use the well-known Stanford Natural Language Inference (SNLI) Corpus (Bowman et al. 2015). We use two models for our analysis on this task: Enhanced Sequential Inference Model (ESIM) (Chen et al. 2017) and Decomposable Attention (DA) (Parikh et al. 2016) with GloVe embeddings (Pennington, Socher, and Manning 2014). The accuracies reported by ESIM is 86.2%, and DA is 85%.

Class Impressions: Modelling natural language inference involves taking in two inputs: premise and hypothesis and deciding the relation between them. The relation can be one amongst entailment, contradiction, and neutral. Following the algorithm in Sec. 3.1, we find both premise and hypothesis together after starting out from a common initial word sequence. Through this, we get a *typical* premise and its corresponding hypothesis for the three output classes (entailment, contradiction, and neutral).

One example per class for the ESIM model is given in Table 5. Unlike sentiment analysis, class impressions for SNLI are not readily interpretable. This is because that while a sentence from the SST corpus can be considered a combination of latent sentiments, the same cannot be assumed of a hypothesis sentence from the SNLI corpus. A statement by itself is not a characteristic hypothesis (or premise). For instance, the SST sentence "You'll probably love it." is a characteristic positive polarity sentence and can be understood to be so by the word 'love'. The same cannot be said for the

[4]https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html#lexicon

Class	Class Impressions
Contradiction	**Hypothesis**: lynched cardinals giraffes lynched lynched a mown extremist natgeo illustration **Premise**: zucchini restrooms swimming golds weekday rock 4 seven named dart
Entailment	**Hypothesis**: civilization va physical supersonic prohibits biathlon body land muffler mobility **Premise**: gecko robed abroad teetotalers blonds pluggling sprinter speeds corks dogtrack
Neutral	**Hypothesis**: porters festivals fluent a playgrounds ratatouille buttercups horseback popularity waist **Premise**: bowler teaspoons group tourism tourism spiritual physical physical person

Table 5: Class Impressions for ESIM model trained for the Natural Language Inference Task

Class Type: Entailment→Neutral Original Accuracy: (ESIM: 91%, DA: 90.3%)				
Data-Inputs	**Data-Type**	**Trigger**	**ESIM**	**DA**
Hypothesis and Premise	Data-Based	nobody	0.06	0.18
		whatsoever	0.6	43
		cats	0.69	0.7
	Data-Free	nobody	0.06	0.18
		no	0.1	2
		mars	0.1	0.3
Hypothesis-Only	Data-Free	monkeys	0.7	0.54
		zebras	0.5	0.39
		cats	0.69	0.7
Class Type: Neutral→Contradiction Original Accuracy: (ESIM: 88%, DA: 80%)				
Data-Inputs	**Data-Type**	**Trigger**	**ESIM**	**DA**
Hypothesis and Premise	Data-Based	shark	18	28
		moon	17	13
		spacecraft	12	8.4
	Data-Free	skydiving	14	20
		orangutan	12	75
		spacecraft	12	8.4
Hypothesis-Only	Data-Free	sleep	11	19
		drowning	15	29
		spacecraft	12	8.4
Class Type: Contradiction→Entailment Original Accuracy: (ESIM: 79%, DA: 85%)				
Data-Inputs	**Data-Type**	**Trigger**	**ESIM**	**DA**
Hypothesis and Premise	Data-Based	expert	64	73
		siblings	66	68
		championship	65	74
	Data-Free	inanimate	67	82
		final	66	68
		championships	68	85
Hypothesis-Only	Data-Free	humans	70	79
		semifinals	68	74
		championship	65	74

Table 6: We prepend a single word (Trigger) to SNLI hypotheses. We display the top 3 triggers created using both Validation set and Class Impressions for ESIM and show their performance on the DA. The original accuracies are mentioned in brackets.

Class	Class Impressions
Paraphase Detected	**Sentence 1**:nintendo daredevil bamba bamba the the lakers dodgers weekend rhapsody seahawks **Sentence 2**: nintendo multiplayer shawnee dodgers anthem netball the olympics soundtrack overture martial
Paraphase Detected	**Sentence 1**: mon submitted icus submit arboretum templar desires them requirements kum **Sentence 2**: lection rahul organizers postgraduate qualifying your exercises signifies its them
No Paraphase Detected	**Sentence 1**: b 617 matrices dhabi ein wm spelt rox a proportional alamo swap **Sentence 2**: drilled traced 03 02 said mattered million 0% 50% corporations a a
No Paraphase Detected	**Sentence 1**: cw an hung kanda singapore tribu chun mid 199798 nies bula latvia **Sentence 2**: came tempered paced times than an saying say shone say s copp

Table 7: Class Impressions for ALBERT model trained for the Microsoft Research Paraphrase Corpus

SNLI premise sentence "An older and younger man smiling." SNLI class impressions give us a glance into a model's learnt deep manifold representation of premise-hypothesis pair. They are generally far away from the training data. Strong priors about the natural training distribution might be needed to make them closer to the training data, . We leave this task for future investigation.

UAT: After obtaining a batch of class impressions from the previous step, we craft the universal adversarial triggers. A comparison of the results for UATs generated using our method, and those of (Wallace et al. 2019) are given in Table 6. As can be seen, we achieve comparable results as (Wallace et al. 2019). A single word trigger is able to reduce the accuracy of the entailment class from 90.3% to 0.06%.

Hypothesis Only UATs: Several recent research studies have indicated that the annotation protocol for SNLI leaves artefacts in the dataset such that by giving just hypothesis, one can obtain 67% accuracy (Gururangan et al. 2018; Poliak et al. 2018). Following that line of study, we generate only the hypothesis class impressions using the CIG algorithm. Then, we generate triggers over the hypothesis-only generated class impressions. Table 6 notes the results for the hypothesis-only attacks. We find that hypothesis-only triggers perform equivalently to hypothesis and premise attacks. This provides further proof that there are many biases in the SNLI dataset and more importantly, the models are using those biases as class representations and adversarial triggers actively exploit these (§5).

Transfer of Mined UATs to Other Models: To determine how the triggers mined from one model transfer to another, we test both data-based and our data-free triggers generated using the ESIM model on the DA model. Table 6 shows the results. We check the transfer attack performance in two cases: where both hypothesis and premise are given and where only the hypothesis is given. It can be seen that even though both the models are architecturally very different, the triggers transfer remarkably well for both cases. For instance, for the entailment class, the original and transfer attack accuracy drops are comparable. It is also noteworthy that our results are equivalent to (Wallace et al. 2019) even for transfer attacks.

4.3 Paraphrase Identification

For paraphrase identification, we use the Microsoft Research Paraphrase Corpus (MRPC) (Dolan and Brockett 2005). Paraphrase identification is the task of identifying whether two sentences are semantically equivalent. We use the ALBERT model (Lan et al. 2020) for the task. It reports an accuracy of 89.9% over this.

Class Impressions: Similar to natural language inference, here, the models require two input sentences. The task of the model is to identify whether the two sentences are semantically the same. The class impressions generated on the ALBERT model are given in Table 7. We find that similar to the SNLI corpus, the MRPC class impressions are not readily interpretable. For specific examples like the first example in the table, we find that sometimes words related to one topic occur as class impressions. Words like 'nintendo' and 'daredevil' in sentence one and 'multiplayer' and 'anthem' often

occur in the context of multiplayer digital games. We should have got similar class impressions in an ideal scenario for sentences 1 and 2 for actual paraphrases. However, we find that the model considers even those sentence pairs (example 2) as paraphrases that have zero vocabulary or topic overlap. This indicates that the model is performing a similarity match in the high dimensional data manifold. We do some analysis for this in Sec. 5. We leave the further investigation of this for future work.

UAT: Table 8 notes the performance of 3 word data-free adversarial triggers generated using MINIMAL. As can be seen, the mined artefacts reduce the accuracy for both classes by more than 70%.

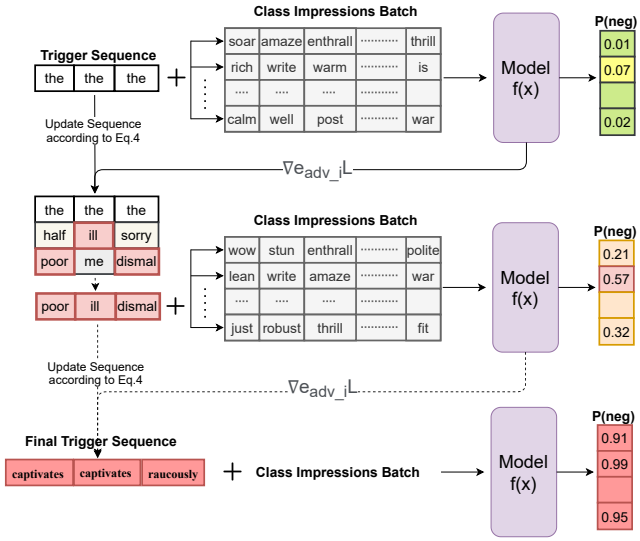

Figure 3: Iterative Universal Trigger Generation (UTG) Algorithm.

5 Analyzing the Class Impressions

We futher analyze class impressions and their relationship with universal adversarial triggers. Specifically, we try to answer these questions: which words get selected as class impressions, why are we able to find universal adversarial triggers from a batch of class impressions and no train data distribution is required? We also try to relate it to the observation made by (Gururangan et al. 2018; Poliak et al. 2018), which ranked the dataset artefact words by calculating their pointwise-mutual information (PMI) values for each class. We further show that the trigger words align very well with dataset artefacts.

Class Impression Words: For analyzing why certain words are selected as representatives of a particular class, we find the discriminative power of each word by calculating its entropy. Concretely, we calculate entropy of the random variable $Y|X$ where Y denotes a model class and X denotes the word level feature. Formally, we compute:

$$\mathbb{H}(Y|X) = -\sum_{k=1}^{K} p(Y=k|X)\log_2 p(Y=k|X) \quad (5)$$

Type	Direction	Trigger	Acc. Before	Acc. After
Data-free	P → N	insisting sacrificing either	95	45
Data-free	N → P	waistband interests stomped	80.9	61.6

Table 8: Accuracy drop for the ALBERT paraphrase identification model after prepending 3-word adversarial triggers generated using MINIMAL.

Stanford Sentiment Treebank			
Positive	**%**	**Negative**	**%**
beautifully	99.97	dull	99.99
wonderful	99.95	worst	99.99
enjoyable	99.94	suffers	99.98
engrossing	99.94	stupid	99.98
charming	99.89	unfunny	99.97
Impression Average	73.89	Impression Average	77.97

Table 9: PMI percentiles for sample class impression words and their average

Microsoft Research Paraphrase Corpus			
Paraphrase	**%**	**Non-Paraphrase**	**%**
experts	99.89	biological	99.91
such	99.84	important	99.39
only	99.67	drug	99.92
due	99.65	case	98.91
said	99.57	among	98.73
Impression Average	77.23	Impression Average	81.89

Table 10: PMI percentiles for sample class impression words and their average

Stanford Natural language Inference					
Contradiction	**%**	**Entailment**	**%**	**Neutral**	**%**
naked	99.99	human	99.91	about	99.73
sleeping	99.97	athletic	99.73	treasure	99.06
tv	99.96	martial	99.71	headed	99.05
asleep	99.96	clothes	99.53	school	98.87
eats	99.93	aquatic	99.38	league	98.83
Average	67.89	Average	70.89	Average	68.97

Table 11: PMI percentiles for sample class impression words and their average

Ground Truth→Attacked Target	Trigger	ESIM
Entailment→Neutral Accuracy:88%	beatboxing	77
	insects	68
	reclining	83
Entailment→Contradiction Accuracy:79%	qualities	70
	coexist	71
	stressful	70
Neutral→Contradiction Accuracy: 79%	disoriented	69
	arousing	67
	championship	65
Neutral→Entailment Accuracy: 91%	championship	0.1
	semifinals	0.9
	aunts	0.5
Contradiction→Entailment Accuracy: 91%	ballet	5
	nap	2
	olives	9
Contradiction→Neutral Accuracy: 88%	nap	14
	hubble	21
	snakes	9

Table 12: We prepend a single word (trigger) to SNLI hypotheses. We take the first word from all ground truth class impressions and evaluate them on class impressions of the target class. We then choose the top 4 and show their validation performance for the target class.

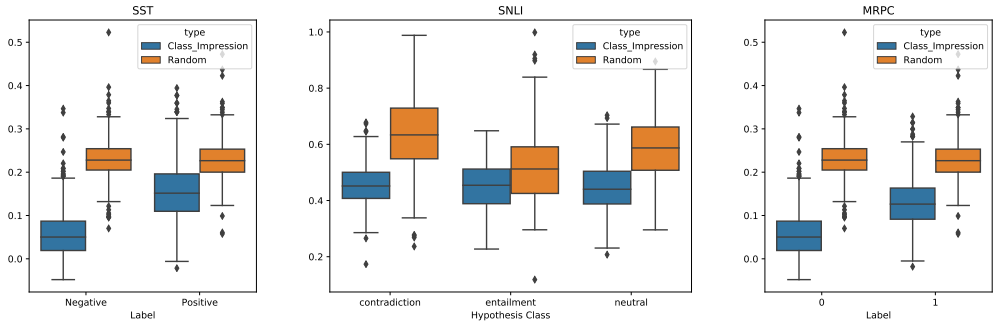

Figure 4: Mean Entropy of class impression words and 350 words randomly selected from the SST, SNLI, and MRPC dataset vocabularies.

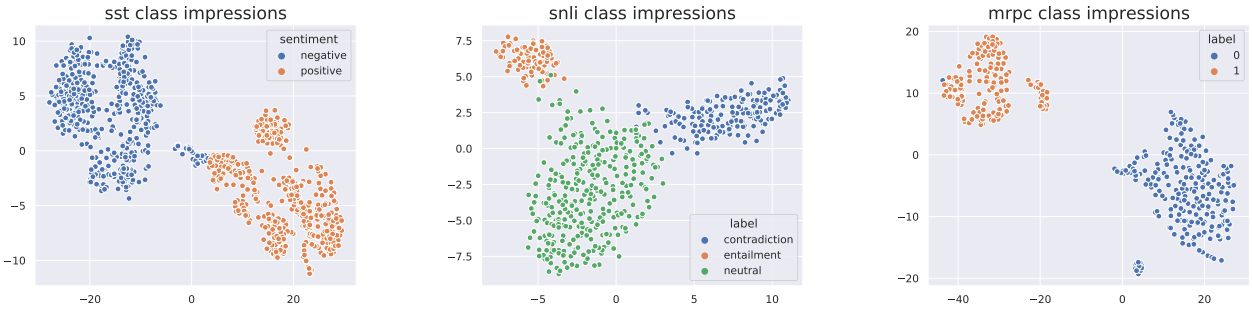

Figure 5: t-SNE plots for SST, SNLI, and MRPC class impression words. The words from the different class impressions form different distinct clusters depending on its class for all three datasets. The clusters are shown in different colors based on their classes.

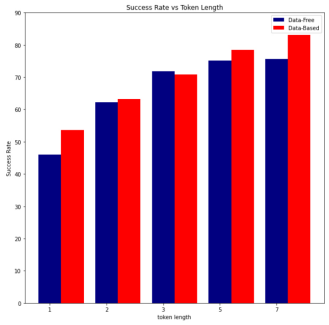

Figure 6: Attack success rate as a function of trigger length

for the class impression words and we compare them with randomly chosen words from the model vocabulary. Fig. 4 shows the results for SST, SNLI, and MRPC datasets. Interestingly, we find that the words which form class impressions are low entropy features. These words are much more discriminative than other randomly sampled words for all three datasets. This is further reinforced by Fig. 5 where we show t-SNE plots for all the datasets . They show that words from different class impressions form distinct clusters.

Fig. 4 shows that CIG algorithm selects low entropy features as representatives of different classes. However, it does not show the class-preference of these low entropy word-features. We hypothesize that those words become representatives of a particular class with a higher PMIs with respect to that class. In order to show this, we calculate PMI values of class representatives for each class and note that class representatives have a higher PMI for their own class than other classes. Formally, we compute:

$$PMI(word, class) = log \frac{p(word, class)}{p(word, .)p(., class)} \quad (6)$$

We use add-10 smoothing for calculating this. We then group each class impression word based on its target class and report their PMI percentile. We show the results in Tables 9-11. It can be seen that class representatives have very high PMI percentiles. Previous studies have characterized high PMI words as *dataset artefacts* (Gururangan et al. 2018; Poliak et al. 2018). Wallace et al. (2019) have also shown that universal adversarial triggers have a high overlap with these dataset artefacts and consequently have high PMI values. Since we observe that class representatives too have high PMI values, we hypothesize that they could act as good adversarial triggers.

Following this, we postulate that adding class impression words of one class to a real example of another class should change the prediction of that example. For validating this, we conduct an experiment where we take words from class impressions of class c_i and prepend them to real examples of class c_j. Table 12 shows the results of the experiment over SNLI dataset. As can be seen, the results are very promising.

We observe that the class which was more adversarially unsecure (*Entailment*$>_{adv-unsecure}$*Contradiction*) has bet-

ter class impression words. These words, when added to examples of other classes, produce more successful perturbations. For e.g., when entailment words are added to contradiction examples, they reduce the accuracy from 91% to less than 10%. On the other hand, contradiction was adversarially more secure, and hence there is no appreciable reduction in the accuracy of any other class upon adding the contradiction class impression words[5]. This result can potentially help dataset designers design more secure datasets on which the model-makers can train adversarially robust models.

The above analysis shows that we can get class-impressions and adversarial triggers from dataset itself by computing entropy and PMI values. Moreover, our experiments in Sec. 4 show that one can equivalently mine models to get class impressions and adversarial triggers. Therefore, we conclude that we can craft both class impressions and adversarial triggers given either dataset or a well-trained model (*i.e.*, the one which can model training data distribution well). Further, the models represent their classes with dataset artefacts. These artefacts are also responsible for making them adversarially unsecure. The lesser the dataset artefacts in a class, the lesser is a trained model's representative capacity for that class, and the more is the model's adversarial robustness for that class. We would like to further develop on these initial results to better dataset design protocols in future work.

6 Conclusion and Future Work

This paper presents a novel data-free approach, MINIMAL to mine natural language processing models for input-agnostic (universal) adversarial triggers. Our setting is more natural, which assumes an attacker does not have access to training data but only the trained model. Therefore, existing data-dependent adversarial trigger generation techniques are unrealistic in practice. On the other hand, our method is data-free and achieves comparable performance to data-based adversarial trigger generation methods. We also show that the triggers generated by our algorithm transfer remarkably well to different models and word embeddings. We achieve this by developing a combination of model inversion and adversarial trigger generation attacks. Finally, we show that low entropy word-level features occur as adversarial triggers and hence one can equivalently mine either a model or a dataset for these triggers.

We conduct our analysis on word-level triggers and class impressions based model inversion. While this analysis leads to crucial insights into dataset design and adversarial trigger crafting techniques, it can be extended to multi-word contextual analysis. This will also potentially lead to better dataset design protocols. We are actively engaged in this line of research. Further, another research focus can be to generate natural-looking class impressions and, consequently adversarial triggers.

[5]We find similar results on the MRPC dataset. We did not do these experiments for the SST dataset since SST class impression words are construct-relevant words and hence are bound to change sentiment scores while the same is not true for the other two datasets.

References

Behjati, M.; Moosavi-Dezfooli, S.; Baghshah, M. S.; and Frossard, P. 2019. Universal Adversarial Attacks on Text Classifiers. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, United Kingdom, May 12-17, 2019*, 7345–7349. IEEE.

Bowman, S. R.; Angeli, G.; Potts, C.; and Manning, C. D. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 632–642. Lisbon, Portugal: Association for Computational Linguistics.

Cambria, E.; Schuller, B.; Xia, Y.; and Havasi, C. 2013. New avenues in opinion mining and sentiment analysis. *IEEE Intelligent systems*, 28(2): 15–21.

Chen, Q.; Zhu, X.; Ling, Z.-H.; Wei, S.; Jiang, H.; and Inkpen, D. 2017. Enhanced LSTM for Natural Language Inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1657–1668. Vancouver, Canada: Association for Computational Linguistics.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 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)*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

Dolan, W. B.; and Brockett, C. 2005. Automatically Constructing a Corpus of Sentential Paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Ebrahimi, J.; Rao, A.; Lowd, D.; and Dou, D. 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)*, 31–36. Melbourne, Australia: Association for Computational Linguistics.

Fredrikson, M.; Jha, S.; and Ristenpart, T. 2015. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 1322–1333.

Google. 2021. The Google Natural Language API. https://cloud.google.com/natural-language#natural-language-api-demo.

Graves, A.; and Schmidhuber, J. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural networks*, 18(5-6): 602–610.

Gururangan, S.; Swayamdipta, S.; Levy, O.; Schwartz, R.; Bowman, S.; and Smith, N. A. 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, Volume 2 (Short Papers)*, 107–112. New Orleans, Louisiana: Association for Computational Linguistics.

Huan, Z.; Wang, Y.; Zhang, X.; Shang, L.; Fu, C.; and Zhou, J. 2020. Data-free adversarial perturbations for practical black-box attack. In *Pacific-Asia conference on knowledge discovery and data mining*, 127–138. Springer.

Khrulkov, V.; and Oseledets, I. V. 2018. Art of Singular Vectors and Universal Adversarial Perturbations. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 8562–8570. IEEE Computer Society.

Kumar, Y.; Aggarwal, S.; Mahata, D.; Shah, R. R.; Kumaraguru, P.; and Zimmermann, R. 2019. Get IT Scored Using AutoSAS -

An Automated System for Scoring Short Answers. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, 9662–9669. AAAI Press.

Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; and Soricut, R. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Li, J.; Ji, R.; Liu, H.; Hong, X.; Gao, Y.; and Tian, Q. 2019. Universal Perturbation Attack Against Image Retrieval. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, 4898–4907. IEEE.

Meng, R.; Zhao, S.; Han, S.; He, D.; Brusilovsky, P.; and Chi, Y. 2017. Deep Keyphrase Generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 582–592. Vancouver, Canada: Association for Computational Linguistics.

Micaelli, P.; and Storkey, A. J. 2019. Zero-shot Knowledge Transfer via Adversarial Belief Matching. In Wallach, H. M.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E. B.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 9547–9557.

Michel, P.; Li, X.; Neubig, G.; and Pino, J. 2019. On Evaluation of Adversarial Perturbations for Sequence-to-Sequence Models. 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)*, 3103–3114. Minneapolis, Minnesota: Association for Computational Linguistics.

Mikolov, T.; Grave, E.; Bojanowski, P.; Puhrsch, C.; and Joulin, A. 2018. Advances in Pre-Training Distributed Word Representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA).

Moosavi-Dezfooli, S.; Fawzi, A.; Fawzi, O.; and Frossard, P. 2017. Universal Adversarial Perturbations. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 86–94. IEEE Computer Society.

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

Mopuri, K. R.; Garg, U.; and Radhakrishnan, V. B. 2017. Fast Feature Fool: A data independent approach to universal adversarial perturbations. In *British Machine Vision Conference 2017, BMVC 2017, London, UK, September 4-7, 2017*. BMVA Press.

Mopuri, K. R.; Uppala, P. K.; and Babu, R. V. 2018. Ask, acquire, and attack: Data-free uap generation using class impressions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 19–34.

Nayak, G. K.; Mopuri, K. R.; Shaj, V.; Radhakrishnan, V. B.; and Chakraborty, A. 2019. Zero-Shot Knowledge Distillation in Deep Networks. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, 4743–4751. PMLR.

Parikh, A.; Täckström, O.; Das, D.; and Uszkoreit, J. 2016. A Decomposable Attention Model for Natural Language Inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2249–2255. Austin, Texas: Association for Computational Linguistics.

Pennington, J.; Socher, R.; and Manning, C. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543. Doha, Qatar: Association for Computational Linguistics.

Peters, M.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2227–2237. New Orleans, Louisiana: Association for Computational Linguistics.

Poliak, A.; Naradowsky, J.; Haldar, A.; Rudinger, R.; and Van Durme, B. 2018. Hypothesis Only Baselines in Natural Language Inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, 180–191. New Orleans, Louisiana: Association for Computational Linguistics.

Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C. D.; Ng, A.; and Potts, C. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 1631–1642. Seattle, Washington, USA: Association for Computational Linguistics.

Song, L.; Yu, X.; Peng, H.-T.; and Narasimhan, K. 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*, 3724–3733. Online: Association for Computational Linguistics.

Tramèr, F.; Zhang, F.; Juels, A.; Reiter, M. K.; and Ristenpart, T. 2016. Stealing machine learning models via prediction apis. In *25th {USENIX} Security Symposium ({USENIX} Security 16)*, 601–618.

Wallace, E.; Feng, S.; Kandpal, N.; Gardner, M.; and Singh, S. 2019. Universal Adversarial Triggers for Attacking and Analyzing NLP. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics.

Xiong, C.; Zhong, V.; and Socher, R. 2017. Dynamic Coattention Networks For Question Answering. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Zhang, C.; Benz, P.; Lin, C.; Karjauv, A.; Wu, J.; and Kweon, I. S. 2021. A survey on universal adversarial attack. *arXiv preprint arXiv:2103.01498*.

Zhang, L.; Wang, S.; and Liu, B. 2018. Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4): e1253.