Universal Adversarial Attacks with Natural Triggers for Text Classification

Liwei Song§ Xinwei Yu§ Hsuan-Tung Peng§ Karthik Narasimhan
Princeton University
di{liweis, xinweiy, hsuantungp, karthikn}@princeton.edu

Abstract

Recent work has demonstrated the vulnerability of modern text classifiers to universal adversarial attacks, which are input-agnostic sequence of words added to any input instance (Wallace et al., 2019; Behjati et al., 2019). Despite being highly successful, the word sequences produced in these attacks are often unnatural, do not carry much semantic meaning, and can be easily distinguished from natural text. In this paper, we develop adversarial attacks that appear closer to natural English phrases and yet confuse classification systems when added to benign inputs. To achieve this, we leverage an adversarially regularized autoencoder (ARAE) (Zhao et al., 2018a) to generate triggers and propose a gradient-based search method to output natural text that fools a target classifier. Experiments on two different classification tasks demonstrate the effectiveness of our attacks while also being less identifiable than previous approaches on three simple detection metrics.1

1 Introduction

In recent times, adversarial attacks have demonstrated significant success in bringing down the performance of modern deep learning methods (Szegedy et al., 2014). An adversary can lightly perturb inputs to appear unchanged to humans but induce incorrect predictions by neural networks. Adversarial attacks started out in computer vision (Goodfellow et al., 2015; Carlini and Wagner, 2017) and have been recently explored in several natural language processing (NLP) domains (Jia and Liang, 2017; Ebrahimi et al., 2018; Alzantot et al., 2018; Zhao et al., 2018b).

Universal adversarial attacks are a special subclass of methods where the same attack perturbation can be applied to any input to the target classifier. These attacks, being input-agnostic, point to more serious shortcomings in trained models and do not require regeneration for each input. While Moosavi-Dezfooli et al. (2017) designed some of the first universal attacks for image classification, Wallace et al. (2019) and Behjati et al. (2019) have recently demonstrated successful universal adversarial attacks for NLP models. However, one limitation of their methods is that the generated attack sequences are often meaningless and irregular text (e.g., “zoning tapping fiennes” from Wallace et al. (2019)). While human readers can easily identify them as unnatural, we also find that simple heuristic methods are sufficient to spot such attacks. For instance, the above attack trigger for sentiment analysis has an average word frequency of around 14 compared to 6.8k for benign inputs in the Stanford Sentiment Treebank (SST) (Socher et al., 2013).

In this paper, we focus on designing natural attack triggers by utilizing a generative model of text. Particularly, we use an adversarially regularized autoencoder (ARAE) (Zhao et al., 2018a), which consists of an autoencoder and a generative adversarial network (GAN), and generates natural text using input noise vectors. This enables us to develop a gradient-based search over the noise vector space for triggers with a good attack performance. Our method, which we call Natural Universal Trigger Search (NUTS), uses projected gradient descent with L2 regularization to avoid using out-of-distribution noise vectors and maintain the naturalness of text generated.

We demonstrate the success of our attacks on two different classification tasks — sentiment analysis and natural language inference (NLI). For instance, the phrase ‘she might not’, generated by our approach, brings down the accuracy of a classi-
fier trained on the Stanford NLI corpus (Bowman et al., 2015) to a mere 1% on entailment decisions. Furthermore, we show that our attack text appears more natural than prior approaches according to three different measures – average word frequency, loss under the GPT-2 language model (Radford et al., 2019), and errors identified by two online grammar checking tools (scr; che). For example, on attacks for the sentiment analysis task, the Scribens grammar checker reports 15.63% errors per word for Wallace et al. (2019), and only 9.38% errors per word for our approach.

2 Related Work

Input-dependent attacks These attacks generate specific triggers for each different input to a classifier. Jia and Liang (2017) fool reading comprehension systems by adding a single distractor sentence to the input paragraph. Ebrahimi et al. (2018) replace words of benign texts with tokens whose word embeddings are close to that of the original words using the direction of the gradient of the model’s loss, causing a drop in performance. Similarly, Alzantot et al. (2018) propose a word-replacing attack method based on genetic algorithms. Zhao et al. (2018b) add adversarial perturbations to the latent embeddings of original text and leverage a text generation model to construct the final attack text.

Universal attacks Universal adversarial attacks are input-agnostic and hence, word-replacing and embedding-perturbing approaches are not applicable. Wallace et al. (2019) and Behjati et al. (2019) concurrently proposed to generate sequences of words that can be added to any input text and fool a target NLP model. Both papers perform gradient-guided searches over the space of word embeddings to choose optimal attack triggers. However, in both cases, the generated attack word sequences are often meaningless, and can be easily detected by a semantic checking process (as we show in Section 4). In contrast, our goal is to generate attack triggers that appear like natural phrases and retain semantic meaning.

GANs for generating attacks Generative adversarial networks (GANs) have also been explored for adversarial attacks, particularly in computer vision. Xiao et al. (2018) train a generator to adversarially perturb input images to fool both the target image classifier and the discriminator. Poursaeed et al. (2018) consider both input-dependent and universal attacks in the image domain by training a generator to fool the target model without using the discriminator. Song et al. (2018) use a standard Auxiliary Classifier GAN to synthesize images, which have different predictions between the auxiliary classifier and the target classifier. Zhao et al. (2018b) leverage a GAN to manipulate an input’s latent vector and separately train an inverter to reconstruct an adversarial example from that perturbed latent vector. We use a GAN-based model (ARAE) for generating attacks for models operating over text.

3 Universal Adversarial Attacks with Natural Triggers

We build upon the universal adversarial attacks proposed by Wallace et al. (2019). To enable natural attack triggers, we use a generative model which produces text using a continuous vector input, and perform a gradient-guided search over this input space. The resulting trigger, which is added to benign text inputs, is optimized so as to maximally increase the loss under the target classification model.

Problem Formulation Consider a pre-trained text classifier $F$ to be attacked. We are given a set of benign input sequences $\{x\}$ with the same ground truth label $y$ and the classifier has been trained to predict $F(x) = y$. Our goal is to find a single input-agnostic trigger, $t$ that when concatenated\(^2 \) with any benign input, causes $F$ to perform an incorrect classification, i.e., $F([t; x]) \neq y$, where $;$ represents concatenation. In addition, we also need to ensure the trigger $t$ is natural fluent text.

Attack trigger generation To ensure the trigger is natural, fluent and carries semantic meaning, we use a pre-trained adversarially regularized autoencoder (ARAE) (Zhao et al., 2018a) (details in Section 4). The ARAE consists of an encoder-decoder structure and a generative adversarial network (GAN) (Goodfellow et al., 2014). The input to the ARAE is a standard Gaussian noise vector $n$, which is first mapped to a latent vector $z$ by the generator. Then the ARAE decoder uses this $z$ to generate a sequence of words – in our case, the trigger $t$. This trigger is then concatenated with a set of benign texts $\{x\}$ to get full attack texts $\{x'\}$.

\(^2\)We follow Wallace et al. (2019) in adding the triggers in front of the benign text.
Figure 1: Overview of our gradient-based attack using an ARAE model. Based on the gradient of the target model’s loss function, we iteratively update the noise vector \( n \) with small perturbation to obtain successful and natural attack triggers.

The overall process can be formulated as follows:

\[
\begin{align*}
    z &= \text{GENERATOR}(n), \\
    t &= \text{DECODER}(z), \\
    x' &= [t; x].
\end{align*}
\]

(1)

We then pass each \( x' \) into the target classifier and compute the gradient of the classifier’s loss with respect to the noise vector, \( \nabla_n \mathcal{L}(F(x'), y) \). Backpropagating through the decoder is not straightforward since it produces discrete symbols. Hence, we use a reparameterization trick similar to the trick in Gumbel softmax (Jang et al., 2017) to sample words from the output vocabulary of ARAE model as a one-hot encoding of triggers, while allowing gradient backpropagation. Figure 1 provides an overview of our attack algorithm, which we call Natural Universal Trigger Search (NUTS).

### Ensuring natural triggers

In the ARAE model, the original noise vector \( n_0 \) is sampled from a standard multi-variant Gaussian distribution. While we can change this noise vector to produce different outputs, simple gradient search may veer significantly off-course and lead to bad generations. To prevent this, following some adversarial attack literature (Goodfellow et al., 2015; Carlini and Wagner, 2017), we use projected gradient descent with an \( l_2 \) norm constraint to ensure the noise vector \( n \) is always within a limited ball around the original noise \( n_0 \). We iteratively update \( n \) as:

\[
n_{t+1} = \Pi_{\mathcal{B}_\epsilon(n_0)}[n_t + \eta \nabla_n \mathcal{L}(F(x'), y)],
\]

where \( \Pi_{\mathcal{B}_\epsilon(n_0)} \) represents the projection operator with the \( l_2 \) norm constraint \( \mathcal{B}_\epsilon(n_0) = \{ n \mid \| n - n_0 \|_2 \leq \epsilon \} \). We try different settings of attack steps, \( \epsilon \) and \( \eta \), selecting the value based on the quality of output triggers. In our experiments, we use 1000 attack steps with \( \epsilon = 10 \) and \( \eta = 1000 \).

### Final trigger selection

Since the generation process is not deterministic, we initialize multiple independent noise vectors and perform our updates (2) to obtain many candidate triggers. Then, we re-rank the triggers to balance both target classifier accuracy \( m_1 \) (lower is better) and naturalness in terms of the average per-token cross-entropy under GPT-2, \( m_2 \) (lower is better). In our experiments, we select the trigger with the minimum value of overall performance score, which is defined as \( m_1 + \lambda m_2 \), as the output of searching algorithm. We select \( \lambda = 0.05 \) to balance the difference in scales of \( m_1 \) and \( m_2 \).

### 4 Experiments

We demonstrate our attack algorithm on two different tasks – sentiment analysis and natural language inference. We employ the model of Wallace et al. (2019) as a baseline and use the same datasets and target classifiers for fair comparison. For the text generator, we use an ARAE model pre-trained on the 1 Billion Word dataset (Chelba et al., 2014).\(^3\) For both our attack (NUTS) and the baseline, we limit the vocabulary of attack trigger words to the overlap of the classifier and ARAE vocabularies.

#### Defense metrics

We also employ three simple defense metrics to measure the naturalness of attacks:

1. **Word frequency**: The average frequency of words in the attack trigger, computed using empirical estimates from the training set of the target classifier.
2. **Language model loss**: The average per-token cross-entropy loss under a pre-trained language model – GPT-2 (Radford et al., 2019).
3. **Automatic grammar checkers**: We calculate the average number of errors in the attack sequences using two online grammar checkers – Scribens (scr) and Chegg Writing (che).

#### 4.1 Sentiment Analysis

**Setup** The target classifier we consider uses word2vec embeddings (Mikolov et al., 2013) and

\(^3\)https://github.com/jakezhaojb/ARAE
Table 1: Universal attack results on both the Stanford Sentiment Treebank (SST) classifier and the Stanford Natural Language Inference (SNLI) classifier. For SST, “+” and “−” represent test sentences with positive sentiment and negative sentiment, respectively. For SNLI, “+”, “0”, and “−” represent test sentence pairs with entailment, neutral, and contradiction relations, respectively. We report the model accuracy after adding the attack triggers to benign test data (a lower accuracy means a more successful attack). ‘No trigger’ refers to classifier accuracy without any attack. Compared to the baseline (Wallace et al., 2019), our attack triggers are slightly less successful at reducing classifier accuracy but generate more natural triggers.

| Task | Trigger length | Test data | NUTS (our model) | Baseline (Wallace et al., 2019) |
|------|----------------|-----------|------------------|-------------------------------|
|      | Trigger        | Classifier accuracy | Trigger          | Classifier accuracy          |
| SST  | +              | 88.29%    | why none of      | 88.29%                       |
|      | −              | 82.94%    | natural energy    | 82.94%                       |
|      | +              | 23.60%    | a flat explosion  | 26.35%                       |
|      | −              | 18.46%    | they can deeply  | 27.25%                       |
|      | −              | 10.05%    | restore the empty | 10.05%                       |
| SNLI | +              | 90.95%    | she might not     | 90.95%                       |
|      | 0              | 88.06%    | there is no one  | 88.06%                       |
|      | +              | 54.58%    | he could leave    | 54.58%                       |
|      | +              | 5.02%     | the new state    | 5.02%                        |
|      | +              | 1.82%     | he is hoping to  | 1.82%                        |
|      | +              | 1.02%     | she might not    | 1.02%                        |
|      | +              | 1.02%     | there is no one  | 1.02%                        |
|      | +              | 0.00%     | the new state    | 0.00%                        |
|      | +              | 0.00%     | he is hoping to  | 0.00%                        |
|      | +              | 0.00%     | read the crowd   | 0.00%                        |
|      | +              | 0.00%     | police after     | 0.00%                        |
|      | +              | 3.74%     | the man drowned  | 3.74%                        |
|      | +              | 26.08%    | he is seen after | 26.08%                       |
|      | +              | 5.02%     | the new state    | 5.02%                        |
|      | +              | 1.82%     | he is hoping to  | 1.82%                        |
|      | +              | 1.02%     | she might not    | 1.02%                        |
|      | +              | 0.00%     | the new state    | 0.00%                        |
|      | +              | 0.00%     | he is hoping to  | 0.00%                        |
|      | +              | 0.00%     | read the crowd   | 0.00%                        |
|      | +              | 0.00%     | police after     | 0.00%                        |
|      | +              | 0.00%     | the man drowned  | 0.00%                        |
|      | +              | 0.00%     | he is seen after | 0.00%                        |
|      | +              | 0.00%     | the new state    | 0.00%                        |
|      | +              | 0.00%     | he is hoping to  | 0.00%                        |
|      | +              | 0.00%     | read the crowd   | 0.00%                        |
|      | +              | 0.00%     | police after     | 0.00%                        |
|      | +              | 0.00%     | the man drowned  | 0.00%                        |
|      | +              | 0.00%     | he is seen after | 0.00%                        |

The top half of Table 1 captures the results of both our attack and the baseline (Wallace et al., 2019) with varied sentence lengths. For both positive and negative instances, our method is able to reduce the target classifier’s accuracy significantly, down to 10.05% in the best attack case. Although this is still higher compared to (Wallace et al., 2019)’s attack, we observe that our attacks are much more natural, fluent and readable while the baseline generates rare words (e.g., stamina) and unnatural phrases (e.g. drown soggy timeout).

This is quantitatively portrayed in Figure 2, which shows the difference in statistics between benign text and each attack according to the metrics of word frequency and GPT-2 loss. We see that our generated attacks are much closer in these statistics to the original text inputs than Wallace et al. (2019). Further, as shown in Table 2, these two grammar checkers (scr; che) report 9.38% and 21.88% errors per word on our attack triggers, compared to 15.63% and 28.13% for Wallace et al. (2019).

4.2 Natural Language Inference

Setup For this task, we use the Stanford Natural Language Inference (SNLI) dataset (Bowman et al., 2015) and the Enhanced Sequential Inference
Figure 2: Difference in (a) average word frequency (normalized) and (b) average GPT-2 loss between benign text \((x)\) and different attack triggers \((t)\) (length 5) for SST and SNLI. All statistic differences are computed as \(\text{stat}(x) - \text{stat}(t)\). For SNLI, we observe that our generated attacks have lower GPT-2 loss values than even the original text, leading to a positive delta.

Table 2: Percentage of grammatical errors in the triggers produced by our model (NUTS) and the baseline \((\text{Wallace et al., 2019})\) according to two online grammar checkers – Scribens \((\text{scr})\) and Chegg Writing \((\text{che})\).

| Task | Scribens | Chegg Writing |
|------|----------|---------------|
|      | Ours | Baseline | Ours | Baseline |
| SST  | 9.38% | 15.63% | 21.88% | 28.13% |
| SNLI | 2.08% | 4.17%  | 12.50% | 20.83% |

Results  From Table 1, we see that our attack performs similar to the baseline on entailment and neutral examples. In fact, both attacks successfully decrease the model’s accuracy to almost 0.00% on both entailment and neutral examples with all trigger lengths. On contradiction examples, our best attack brings the model accuracy down to 26.08% while the baseline brings it down to 13.24%. Although our attacks are less successful on contradiction, they are much more natural than the baseline. In Figure 2, our attacks are closer to the word frequency of benign inputs and even achieve a lower GPT-2 loss than the original text. Further, as shown in Table 2, two grammar checkers \((\text{scr}; \text{che})\) report 2.08% and 12.5% errors per word on our attacks, compared to 4.17% and 20.83% for the baseline.

5 Conclusion

In this paper, we develop universal adversarial attacks with natural triggers for text classification models. We leverage the ARAE text generation model and propose a gradient-based approach to search over attack triggers which are fluent and semantically plausible. Experimental results on two different classification tasks validate our approach and show that our model can generate attack triggers that are both successful and natural. Future work can explore better ways to optimally balance attack success and trigger quality, while also investigating more sophisticated ways of detecting attacks.

References

Chegg Writing: Check Your Paper for Free. \url{https://writing.chegg.com/}.

Scribens: Free Online Grammar Checker. \url{https://www.scribens.com/}.

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.

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.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing.

Nicholas Carlini and David Wagner. 2017. Towards evaluating the robustness of neural networks. In IEEE Symposium on Security and Privacy (S&P), pages 39–57.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2014. One billion word benchmark for measuring progress in statistical language modeling. In Fifteenth Annual Conference of the International Speech Communication Association.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced lstm for natural language inference. In ACL (1), pages 1657–1668. Association for Computational Linguistics.
