A Geometry-Inspired Attack for Generating Natural Language Adversarial Examples

Zhao Meng, Roger Wattenhofer
Department of Electrical Engineering and Information Technology
ETH Zurich, Switzerland
{zhmeng, wattenhofer}@ethz.ch

Abstract

Generating adversarial examples for natural language is hard, as natural language consists of discrete symbols, and examples are often of variable lengths. In this paper, we propose a geometry-inspired attack for generating natural language adversarial examples. Our attack generates adversarial examples by iteratively approximating the decision boundary of Deep Neural Networks (DNNs). Experiments on two datasets with two different models show that our attack fools natural language models with high success rates, while only replacing a few words. Human evaluation shows that adversarial examples generated by our attack are hard for humans to recognize. Further experiments show that adversarial training can improve model robustness against our attack.

1 Introduction

Although Deep Neural Networks (DNNs) have been successful in many machine learning applications (Kim, 2014; Rajpurkar et al., 2016; He et al., 2016), researchers have demonstrated that DNNs are remarkably vulnerable to adversarial attacks, which generate adversarial examples by adding small perturbations to the original input (Szegedy et al., 2014; Goodfellow et al., 2015; Nguyen et al., 2015). Adversarial examples are essential as they showcase the limitations of DNN models. Like humans, good DNN models should be robust to small perturbations to inputs. If a DNN model judges two almost identical inputs differently, one must profoundly question the quality of the DNN. As such adversarial examples are more than just a gimmick: they are a proof of the fundamental limitations of a DNN model.

Previous research on adversarial attacks has been largely focused on images, e.g., (Akhtar and Mian, 2018). In this paper, we study how to adversarially attack natural language models. Generating adversarial examples for natural language is fundamentally different from generating adversarial examples for images. Images live in a continuous universe, where one can simply change pixel values. Natural language sentences and words on the other hand are typically discrete. This discrete nature makes it difficult to apply existing attacks from the image domain directly to natural language, as an arbitrary point in the input space is unlikely to correspond to a valid natural language sentence or word. Moreover, inputs of natural language to DNNs are of variable lengths, which further complicates generating adversarial examples for natural language.

Despite these obstacles, researchers have proposed various attacks to generate adversarial examples for natural language. Jia and Liang (2017) manage to fool a DNN model for machine reading by adding sentences to the original texts. Zhao et al. (2018) generate adversarial examples for natural language by using an autoencoder. Ebrahimi et al. (2018) propose a gradient-based attack to generate adversarial examples in the granularity of individual characters. Zhang et al. (2019) generate fluent adversarial examples using Metropolis-Hastings sampling. Ren et al. (2019) combine several heuristics to generate adversarial examples.

However, all these methods do not address the "geometry" of DNNs, which has been shown to be a useful approach in the image domain (Moosavi-Dezfooli et al., 2016; Moosavi-Dezfooli et al., 2019; Modas et al., 2019). In this paper, we propose a geometry-inspired attack for generating natural language

Proceedings of the 28th International Conference on Computational Linguistics, pages 6679–6689
Barcelona, Spain (Online), December 8-13, 2020

adversarial examples. Our attack generates adversarial examples by iteratively approximating the decision boundary of DNNs. We conduct experiments with Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) on two text classification tasks: the IMDB movie review dataset, and AG's News dataset. Experimental results show that our attack fools the models with high success rates while keeping the word replacement rates low. We also conduct a human evaluation, showing that adversarial examples generated by our attack are hard for humans to recognize. Further experiments show that model robustness against our attack can be achieved by adversarial training.

2 Related Work

Despite the success of Deep Neural Networks (DNNs) in many machine learning applications (Kim, 2014; He et al., 2016), researchers have revealed that such models are vulnerable to adversarial attacks, which fool DNN models by adding small perturbations to the original input (Goodfellow et al., 2015). The vulnerability of DNN models poses threats to many applications requiring high-level security. For example, in the image domain, a small error in a self-driving car could lead to life threatening disaster. For natural language, a machine might misunderstand a meaning, coming to a wrong conclusion. Researchers have also shown that a universal trigger could lead a system to generate highly offensive language (Wallace et al., 2019).

Previously, researchers have developed various adversarial attacks for fooling DNN models for images. Goodfellow et al. (2015) propose Fast Gradient Signed Method (FGSM), which aims to maximize the loss of the model with respect to the correct label. Projected Gradient Descent (PGD) (Madry et al., 2018) can be viewed as a multi-step version of FGSM. In each step, PGD generates a perturbation using FGSM, and then projects the perturbed input to an l_∞ ball. While these gradient-based methods are effective, researchers also show that leveraging geometry information of DNNs can be helpful. Moosavi-Dezfooli et al. (2016) and Modas et al. (2019) generate adversarial examples by iteratively approximating the decision boundary of DNNs.

Although many methods have been proposed for generating adversarial examples for images, little attention has been paid to generating adversarial examples for natural language. Generating adversarial examples for natural language is fundamentally different from generating adversarial examples for images. On the one hand, while pixel values of images are continuous, natural language consists of sequences of discrete symbols. Moreover, natural language sentences and words are often of variable lengths. Hence, existing adversarial attacks designed for images cannot be directly applied to natural language.

Despite obstacles, researchers have proposed various methods for generating adversarial examples for natural language. Based on the granularity of adversarial perturbations, adversarial attacks for natural language models can be divided into three categories: character level, word level and sentence level.

2.1 Character Level

Character-level adversarial attacks for natural language models generate adversarial examples by modifying individual characters of the original example. Ebrahimi et al. (2018) propose HotFlip, which uses gradient information to swap, insert, or delete a character in an original example. Li et al. (2019) generate adversarial examples by first selecting important words, and then modifying characters of the selected words.

Although character-level adversarial attacks for natural language are effective, such methods suffer from the problem of perceptibility. Humans are likely to recognize adversarial examples generated by these methods, as changing individual characters of texts often results in invalid words. Furthermore, character-level adversarial attacks are easy to be defended against. Using a simple spell checking tool to preprocess inputs can defend a DNN model against such attacks.

2.2 Word Level

Word-level adversarial attacks generate adversarial examples for natural language by changing words of the original example. Alzantot et al. (2018) propose a genetic attack, in which they replace original words with their synonyms by iteratively applying a genetic algorithm. Zhang et al. (2019) generate adversarial examples for natural language by leveraging Metropolis-Hastings Sampling. Ren et al. (2019) leverage several heuristics to generate word-level adversarial examples. Wallace et al. (2019) propose a universal attack, in which a fixed, input-agnostic sequence of words triggering the model to make a specific prediction is prepended to any example from the dataset. They search such universal triggers by leveraging gradient information.

2.3 Sentence Level

While most researchers focus on character/word-level attacks, some researchers propose to fool DNN models for natural language with sentence-level attacks. Jia and Liang (2017) propose to fool a machine reading model by adding an additional sentence to the original texts. However, their method requires heavy human engineering. Iyyer et al. (2018) generate adversarial examples by rewriting the entire sentence with an encoder-decoder model for syntactically controlled paraphrase generation.

All these methods, however, do not address the geometry of DNNs although such information has been proven useful for generating adversarial examples for images. In this paper, we propose a geometry-inspired word-level adversarial attack for generating natural language adversarial examples. The rest of this paper is organized as follows. Section 3 describes our attack. Section 4 details the experimental settings as well as results. Section 5 gives conclusions and insights for future work.

3 Methodology

Our attack is a white-box attack in that the attacker has access to the architecture and parameters of the victim model. The attack crafts natural language adversarial examples by replacing original words with their synonyms. Specifically, our attack can be divided into two steps: word selection and synonym replacement. In each iteration, the attack first selects a word from the original text, and then replaces the selected word with one of its synonyms to craft an adversarial example. The remainder of this section gives the details of our attack.

3.1 Word Selection Strategy

A crucial step in generating text adversarial examples is to find which word of the original example to replace. We follow previous work by ranking words with their saliency scores (Li et al., 2016a; Li et al., 2016b; Ren et al., 2019). The saliency score of word w_j is obtained by computing the decrease of true class probability after replacing w_j with an out-of-vocabulary word u, embeddings of which are initialized to all zeros during training.

Specifically, we have

$$\boldsymbol{X} = w_0, w_1, \dots, w_j, \dots, w_{N-1} \tag{1}$$

$$\boldsymbol{X}' = w_0, w_1, \dots, u, \dots, w_{N-1} \tag{2}$$

where $\boldsymbol{X}'$ is obtained by replacing word w_j of the original example $\boldsymbol{X}$ with out-of-vocabulary word u. Let y be the ground truth label of original example $\boldsymbol{X}$. The saliency score S_j for word w_j is given by

$$S_j = P(y|\boldsymbol{X}) - P(y|\boldsymbol{X}') \tag{3}$$

A higher saliency score indicates the corresponding word is of more importance for predicting the true class. Hence, the word with the highest saliency score in candidate set $\mathbb{C}$ will be selected for replacement. We build the candidate set $\mathbb{C}$ from words of the original example $\boldsymbol{X}$ and then exclude all out-of-vocabulary words and punctuations.

Algorithm 1 Adversarial Attack

```
input: Example $\boldsymbol{X} = w_0, w_1, \ldots, w_{N-1}$, true label $y$, classifier $f$ with text encoder Encoder and feed forward layer FFNN.
output: Adversarial example $\hat{\boldsymbol{X}}$.
Initialize $\boldsymbol{X}_0 \leftarrow \boldsymbol{X}$, candidate set $\mathbb{C} = \{w_0, w_1, \ldots, w_{K-1}\}$, $i \leftarrow 0$, projections $\mathbb{P} \leftarrow \{\}$, $i \leftarrow 0$.
while $\mathbb{C} \neq \emptyset$ do
    for $w_k \in \mathbb{C}$ do
        $S_k \leftarrow$ WordSaliency($\boldsymbol{X}_i$, $w_k$) // compute word saliency
    end for
    $k^* \leftarrow \text{argmax}_k\, S_k$, where $w_k \in \mathbb{C}$
    $\mathbb{Q}_k^* \leftarrow \{w_{k^*}^0, w_{k^*}^1, \ldots, w_{k^*}^{M_j-1}\}$ // synonym set of $w_{k^*}$
    $\boldsymbol{v}_i \leftarrow \texttt{Encoder}(\boldsymbol{X}_i)$
    $\boldsymbol{b}_i \leftarrow \texttt{DeepFool}(\boldsymbol{v}_i, \texttt{FFNN})$
    $\boldsymbol{r}_i \leftarrow \boldsymbol{b}_i - \boldsymbol{v}_i$
    $\boldsymbol{u}_i \leftarrow \frac{\boldsymbol{r}_i}{||\boldsymbol{r}_i||}$
    for $m = 0$ to $M_j - 1$ do
        Craft $\boldsymbol{X}_i^m$ by replacing $w_{k^*}$ with $w_{k^*}^m$
        $\boldsymbol{v}_i^m \leftarrow \texttt{Encoder}(\boldsymbol{X}_i^m)$
        $\boldsymbol{d}_i^m \leftarrow \boldsymbol{v}_i^m - \boldsymbol{v}_i$
        $\boldsymbol{p}_i^m \leftarrow \texttt{Projection}(\boldsymbol{d}_i^m, \boldsymbol{r}_i)$
        $\mathbb{P} \leftarrow \mathbb{P} \cup \{\boldsymbol{p}_i^m\}$
    end for
    $m^* \leftarrow \text{argmax}_m(\boldsymbol{p}_i^m \cdot \boldsymbol{u}_i)$, where $\boldsymbol{p}_i^m \in \mathbb{P}$
    $\boldsymbol{X}_{i+1} \leftarrow \boldsymbol{X}_i^{m^*}$
    if $f(\boldsymbol{X}_{i+1}) \neq f(\boldsymbol{X})$ then
        $\hat{\boldsymbol{X}} \leftarrow \boldsymbol{X}_{i+1}$
        break
    end if
    $\mathbb{C} \leftarrow \mathbb{C} - \{w_{k^*}\}$
    $i \leftarrow i + 1$
end while
return $\hat{\boldsymbol{X}}$
```

3.2 Synonym Replacement Strategy

Before going into the details of our synonym replacement strategy, we first clarify our assumptions on model architectures. For text classification tasks, a model can be divided into a text encoder `Encoder` and a feed forward layer `FFNN`. Specifically, a text encoder encodes an input $\boldsymbol{X}$ into a fixed-size vector $\boldsymbol{v}$. Choices of such encoders include RNNs, CNNs (Kim, 2014), etc. A feed forward layer then takes the fixed-size vector $\boldsymbol{v}$ as input for classification. A fully connected network followed by a softmax activation layer is common for feed forward layers.

Our attack iterates over the candidate set $\mathbb{C}$ to generate adversarial examples. In each iteration, we first compute word saliency score S_k for each candidate word $w_k \in \mathbb{C}$. We derive the synonym set $\mathbb{Q}_{k^*} = \{w_{k^*}^0, w_{k^*}^1, \ldots, w_{k^*}^{M_j-1}\}$ using `WordNet`[1] for candidate word w_{k^*}, which has the largest saliency score S_{k^*} in the current iteration.

We then use geometric information to select the best synonym of w_{k^*} for replacement. Given a DNN classifier consisting of text encoder `Encoder` and feed forward layer `FFNN`, we first use `Encoder` to compute the text vector $\boldsymbol{v}_i$ of $\boldsymbol{X}_i$, which is the example before replacement at iteration i. We then find the nearest point $\boldsymbol{b}_i$ on the decision boundary of `FFNN` by leveraging the `DeepFool` algorithm (Moosavi-Dezfooli et al., 2016). Next, we compute $\boldsymbol{r}_i$, which originates from text vector $\boldsymbol{v}_i$ to decision boundary

[1] `https://wordnet.princeton.edu/`

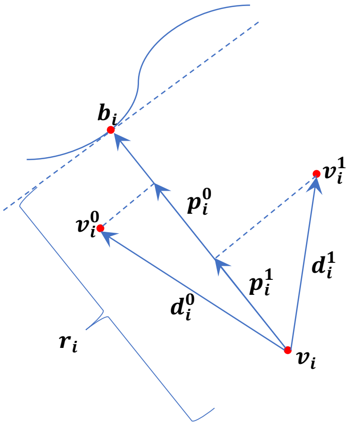


Figure 1: Illustration of iteration i in our attack: $\boldsymbol{v}_i$ is the original text vector. The curved line on top is the decision boundary, with $\boldsymbol{b}_i$ being the closest point on the decision boundary to $\boldsymbol{v}_i$. $\boldsymbol{v}_i^0$ and $\boldsymbol{v}_i^1$ are text vectors obtained by replacing word w_{k^*} with its synonyms $w_{k^*}^0$ and $w_{k^*}^1$, respectively. $\boldsymbol{p}_i^0$ ($\boldsymbol{p}_i^1$) is the projection of $\boldsymbol{d}_i^0$ ($\boldsymbol{d}_i^1$) onto $\boldsymbol{r}_i$. In this iteration, $w_{k^*}^0$ is chosen over $w_{k^*}^1$ as $||\boldsymbol{p}_i^0|| > ||\boldsymbol{p}_i^1||$. We also have $z_i^{max} = ||\boldsymbol{p}_i^0||$ in this example.

point $\boldsymbol{b}_i$.

For each synonym $w_{k^*}^m \in \mathbb{Q}_{k^*}$, example $\boldsymbol{X}_i^m$ is obtained by replacing w_{k^*} with $w_{k^*}^m$. We compute text vector $\boldsymbol{v}_i^m$ by feeding $\boldsymbol{X}_i^m$ into the `Encoder`. We obtain $\boldsymbol{p}_i^m$ by projecting $\boldsymbol{d}_i^m$, which is the vector originating from $\boldsymbol{v}_i$ to $\boldsymbol{v}_i^m$, onto $\boldsymbol{r}_i$. A new example $\boldsymbol{X}_{i+1}$ is crafted by replacing original word w_{k*} with its synonym $w_{k^*}^{m*}$, which corresponds to the largest projection z_i^{max}, where $z_i^{max} = \boldsymbol{p}_i^{m^*} \cdot \boldsymbol{u}_i$, with $\boldsymbol{u}_i$ being the unit direction vector of $\boldsymbol{r}_i$. Our intuition is that a text vector with larger projection on $\boldsymbol{r}_i$ is closer to the decision boundary. We assign $\boldsymbol{X}_i$ to $\boldsymbol{X}_{i+1}$ directly and continue to the next iteration if z_i^{max} is negative (which indicates $\boldsymbol{p}_i^{m^*}$ is in the opposite direction of $\boldsymbol{u}_i$ and $\boldsymbol{r}_i$). Figure 1 illustrates our synonym replacement strategy. The algorithm stops under the condition that the model is fooled or the candidate set $\mathbb{C}$ is exhausted. We give details of our attack in Algorithm 1.

4 Experimental Results

We elaborate our experiments in this section[2]. Section 4.1 details the experimental settings. Section 4.2 describes the results of adversarial attacks. We conduct a human evaluation in Section 4.3 to understand the perceptibility of our adversarial perturbations. Section 4.4 gives the results of adversarial training, which we found can improve the robustness of DNN models against our attack.

4.1 Setup

We describe our experimental setup, including datasets and models in this subsection. We test our attack on two datasets with two different models.

Datasets We conduct our experiments on two English datasets for text classification. Specifically, we have

- **IMDB**[3] (Maas et al., 2011): The IMDB dataset is a large dataset for binary sentiment classification. Each example in the dataset is a movie review. The classification label is `positive`/`negative`. Both labels are equally distributed in the dataset.

- **AG's News**[4]: The AG's News dataset consists of news articles for topic classification. The dataset has four equally distributed labels: `World`, `Sports`, `Business` and `Sci/Tech`.

[2] Our code is available at: `https://github.com/zhaopku/nlp_geometry_attack`

[3] `https://ai.stanford.edu/~amaas/data/sentiment/`

[4] `http://groups.di.unipi.it/~gulli/AG_corpus_of_news_articles.html`

Example	Predictions	Replacements	Distance	True Class Prob
Obviously, most of the budget was put into the dinosaurs, and although there is a fair share of them, there's not nearly enough to **save** (**preserve**) us from our **boredom** (**ennui**). These human characters are only there to scream, run around, and mutter these poorly-written and verbose speeches about survival. And **unfortunately** (**regrettably**), not nearly enough of them get eaten by the dinosaurs. Overall , "planet of the dinosaurs" is not a film I plan on seeing again.	`Negative` → `Positive`	**boredom** → **ennui** **save** → **preserve** **unfortunately** → **regrettably**	0.80 → 0.56 0.56 → 0.25 0.25 → -0.14	84.19% → 76.55% 76.55% → 62.62% 62.62% → 42.93%
Screening as part of a series of funny shorts at the sydney gay and lesbian mardi gras film festival, this film was **definitely** (**unquestionably**) a highlight. The script is **great** (**smashing**) and the direction and acting was terrific. As another posting said, the actors' comedic timing really made this film. Lots of fun.	`Positive` → `Negative`	**great** → **smashing** **definitely** → **unquestionably**	1.56 → 0.01 0.01 → -0.79	96.37% → 50.12% 50.12% → 15.88%
This is the first movie I have watched in ages where I actually ended up fast forwarding through the **tedious** (**wordy**) bits which there are plenty of. Very ordinary movie. I'm glad I missed it at the movies & got a 2 for 1 video deal which included this movie instead.	`Negative` → `Positive`	**tedious** → **wordy**	2.45 → -1.89	99.42% → 40.21%
Ready (**Prepare**) to **bet** (**depend**) on alternative energy? Well, think again when oil prices rise, public interest in alternative energy often does, too. But the logic is evidently escaping wall street.	`Business` → `Sci/Tech`	**bet** → **depend** **ready** → **prepare**	2.65 → 0.34 0.34 → -2.19	99.65% → 68.03% 68.03% → 0.51%
Convicted spammer gets nine years in **slammer** (**jailhouse**) A brother and sister have been convicted of three felony charges of sending thousands of junk e-mails; one of them was sentenced to nine years in prison, the other was fined $ 7,500.	`Sci/Tech` → `Business`	**slammer** → **jailhouse**	2.64 → -0.98	99.77% → 8.67%
Osaka school **killer** (**slayer**) of 8, Yakuza boss executed Yokyo - Mamoru Takuma, convicted for murdering eight children at an Osaka elementary school in 2001, has been executed, informed sources said Tuesday.	`World` → `Sports`	**killer** → **slayer**	1.59 → -1.60	98.20% → 1.60%

Table 1: Adversarial examples from our attack. Irrelevant parts of an example are omitted for simplicity. The first three examples are from the IMDB dataset, and the last three are from AG's News dataset. We use LSTM-based RNN for both datasets. **Green** words are original words, while **red** words are replaced words. **Predictions**: model predictions before and after the attack. **Replacements**: word replacements. **Distance**: changes of distance from text vector to decision boundary. **True Class Prob**: changes of true class probability as original words being replaced. "True Class" refers to the true class of the original example.

Dataset	#Train	#Test	#Classes	Avg. #Words	Max. #Words
IMDB	25,000	25,000	2	258	600
AG's News	120,000	7,600	4	43	248

Table 2: Statistics of datasets. Note that we limit the maximum number of words per example for the IMDB dataset to 600, while we do not limit the maximum number of words for the AG's News dataset.

Model Dataset	CNN	RNN
IMDB	88.49	85.69
AG's News	92.18	91.17

Table 3: Test accuracy (%) of our model on clean examples.

Method	IMDB				AG's News			
	CNN		RNN		CNN		RNN	
	% Replaced$^{\downarrow}$	% Success$^{\uparrow}$	% Replaced$^{\downarrow}$	% Success$^{\uparrow}$	% Replaced$^{\downarrow}$	% Success$^{\uparrow}$	% Replaced$^{\downarrow}$	% Success$^{\uparrow}$
Ren et al. (2019)	3.59	88.95	3.79	84.09	**10.01**	85.95	15.33	79.90
Our Attack	**3.19**	**96.12**	**2.97**	**99.09**	16.33	**86.49**	**14.91**	**87.08**

Table 4: Results of adversarial attacks. **Replaced**: Average word replacement rate. **Success**: Success rate of attack. Larger$^{\uparrow}$ (or lower$^{\downarrow}$) numbers indicate the attack is more efficient.

We list the details of the datasets in table 2. Note that in preprocessing, we limit the maximum number of words to 600 for each example in the IMDB dataset. We do not limit the maximum number of words in the AG's News dataset. Additionally, examples in both datasets are tokenized using NLTK[5]. The average/maximum number of words is computed after preprocessing.

Models We consider two different DNN models to test the effectiveness of our attack. Specifically, we use word-based convolutional neural networks (CNN) and recurrent neural networks (RNN). We use LSTM as the recurrent unit in RNN. A CNN or RNN is a text encoder, which takes as input texts $\boldsymbol{X}$ and outputs a fixed-size vector $\boldsymbol{v}$. A fully connected layer with softmax activation is followed for classification. For both models, we use 100-dimensional GloVe embeddings[6] (Pennington et al., 2014) in our experiments. All hidden layers are 128-dimensional. Table 3 gives the performance of our model on clean examples. These results are comparable to model performance in other studies (Alzantot et al., 2018; Ren et al., 2019), which means that our implementation is fair and that we are ready to investigate performance of our adversarial attacks on these models.

4.2 Adversarial Attacks

We limit the maximum number of word replacements to 50 and 25 for the IMDB dataset and AG's News dataset, respectively. In other words, the algorithm gives up if it still cannot find an adversarial example after the number of words replaced in the original example has exceeded the limit. We report the success rate of our attack on all *correctly classified* examples from the testset to prevent the model performance on clean examples from confounding the attack results. We also report the average word replacement rate for our adversarial examples. A lower word replacement rate makes it harder for humans to distinguish adversarial examples from the original ones.

We compare our attack with Probability Weighted Word Saliency (PWWS) (Ren et al., 2019), which uses a greedy algorithm based on heuristics like word saliency and true class probability. For a fair comparison, the max length of examples is set to 600 for the IMDB dataset. We do not limit the maximum number of words for the AG's News dataset. To facilitate comparison, we obtain the results of PWWS for each dataset by evaluating on 1,000 randomly selected original examples from the testset, while we evaluate our attack on the entire testset.

Table 4 shows the results of our adversarial attacks. As we can see, our attack outperforms PWWS in most of the metrics. For the IMDB dataset, our attack fools the CNN and RNN model with success

[5] `https://www.nltk.org/`

[6] `https://nlp.stanford.edu/projects/glove/`

Dataset	% Accuracy		Similarity		Modified	
	Original	Adversarial	Original	Adversarial	Original	Adversarial
IMDB	92	90	4.13	3.40	2.59	3.14
AG's News	90	81	4.96	3.29	2.18	3.16

Table 5: Human evaluation. **Accuracy**: prediction accuracy of human on the examples. **Similarity**: Given a score of 1-5, judge the similarity of the example text to the original text. **Modified**: Given a score of 1-5, judge the possibility of the example text having been modified by a machine. Higher score indicates higher similarity/possibility.

rates of 96.12% and 99.09%, respectively. Our success rates are higher than the success rates of PWWS (88.95% for CNN, 84.09% for RNN). While reaching higher success rates, our method also has lower word replacement rates than PWWS. The average word replacement rate is 3.19% for CNN model and 2.97% for RNN model, both of which are lower than results from PWWS (3.59% for CNN, 3.79% for RNN).

For AG's News dataset, the success rate of our method is 86.49% for CNN and 87.08% for RNN. The average word replacement rate is 16.33% for CNN and 14.91% for RNN. Our method still outperforms PWWS in the RNN model, where the success rate and average word replacement rate of PWWS are 79.90% and 15.33%, respectively. However, in the CNN model, while having a similar success rate, our model has a higher word replacement rate of 16.33% compared to 10.01% of PWWS.

Compared to attack results of the IMDB dataset, we obtain lower success rates and higher word replacement rates for AG's News dataset. Possible explanations are: (1) fooling a multi-class classifier is harder than fooling a binary classifier, (2) examples of the IMDB dataset are longer than examples of AG's News dataset, and it is easier to generate adversarial examples for longer sequences.

Table 1 gives some adversarial examples generated by our attack. As the attack replaces words from the original example, the true class probability decreases and the resulting text vector moves closer to the decision boundary. A distance smaller than 0 indicates that the text vector has crossed the decision boundary.

4.3 Human Evaluation

We conducted a human evaluation to understand the perceptibility of our adversarial perturbations. For each dataset, we randomly select 100 adversarial examples and the corresponding original examples. We hired workers from Amazon Mechanical Turk[7] to conduct the evaluation. We asked the workers to perform three tasks:

(1) **Accuracy**: Predict the label of an example.

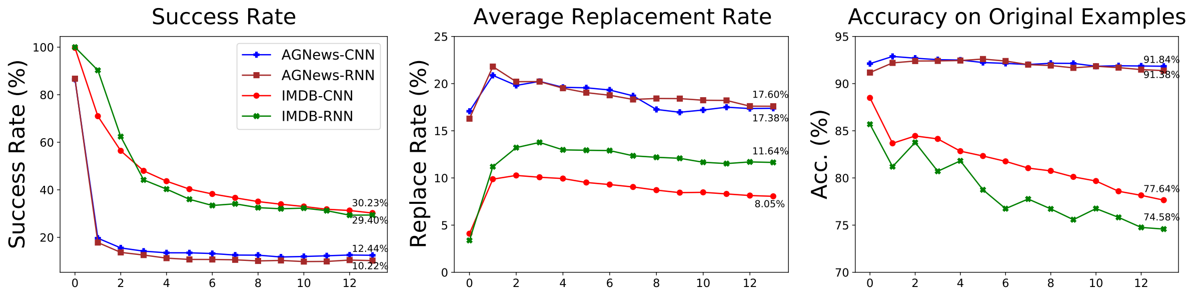
(2) **Similarity**: Judge the similarity of the given example to the original example.

(3) **Modified**: Judge the possibility that some words in the texts having been replaced by a machine.

For the last two tasks, the workers are required to give a score between 1 to 5. A higher score indicates more similarity/higher possibility. For each task, each assignment is shown to five workers. All assignments are randomly shuffled before shown to workers. For task (1), we take the majority of the five predictions as our final label. Note that for the AG's News dataset, we count an example as incorrectly classified if no majority label exists. For tasks (2) and (3), we average scores across workers.

Table 5 shows the results of our human evaluation. For the IMDB dataset, the prediction accuracy on adversarial examples is only 2% lower than the accuracy on original examples. This shows that adversarial examples generated by our attack mostly preserve the content and sentiment of the original examples. The evaluation on similarity and possibility of modifications also shows that the perceived difference between our adversarial examples and the original examples is relatively small. Note that the perturbations are more perceptible on AG's News dataset, which is expected as the average word replacement rate of AG's

[7] https://www.mturk.com/



(a)

(b)

(c)

Figure 2: Results of adversarial training. The x-axis is epoch. Epoch 0 corresponds to models trained only on clean examples. We put y values of the last epoch explicitly on the figures. (a) Success rate of our attack. (b) Average word replacement rate of adversarial examples. (c) Model performance on original examples.

News dataset is higher than that of the IMDB dataset. We also notice that although some examples are not grammarly correct after perturbing, the adversarial examples are still hard for humans to recognize as the word replacement rates are very low.

To better understand human performance in judging similarity of texts, the workers were also asked in task (2) to give the similarity score between two identical original examples (refer to column 4 of Table 5). We see from Table 5 that although the workers were expected to give a score of 5 for identical examples, they gave a score of 4.13 and 4.96 for the IMDB and AG's News dataset, respectively. The score for the IMDB dataset (4.13) is lower than that of the AG's News dataset (4.96). We believe that examples of the IMDB dataset longer than the examples of the AG's News dataset makes it harder for workers to judge whether or not two examples are identical.

4.4 Adversarial Training

We conduct further experiments to validate if robustness against our attack can be achieved by adversarial training. To save time, we do adversarial training by fine-tuning on pretrained models. During adversarial training, the training set is augmented by adversarial examples, which successfully fool the model and are generated in each epoch by perturbing the *correctly classified* examples.

Figure 2 (a) shows that adversarial training helps, as the success rates of the attack gradually drops. Figure 2 (b) demonstrates that the average replacement rates increase as we do adversarial training. Lower success rates and higher word replacement rates show that the models are gaining robustness against our attack by adversarial training. We also notice from Figure 2 (b) that the average word replacement rates of adversarial examples start to decrease after training for some epochs. We believe that the model becomes more robust by first identifying adversarial examples with higher word replacement rates. Hence, the adversarial examples left after some epochs of adversarial training have relatively lower word replacement rates.

Figure 2 (c) shows the model accuracy on clean examples. For the IMDB dataset, adversarial training gradually lowers the model accuracy on clean examples. This is in line with previous image domain research, showing that model robustness is at odds with accuracy (Tsipras et al., 2019). However, we do not observe this phenomenon for the AG's News dataset. This indicates that although adversarial training for texts and images are similar, they are still different in certain aspects.

5 Conclusion and Future Work

In this paper, we propose a geometry-inspired attack for generating natural language adversarial examples. Our attack generates adversarial examples by iteratively approximating the decision boundary of Deep Neural Networks. Experiments on two text classification tasks with two models show that our attack reaches high success rates while keeping low word replacement rates. Human evaluation shows that adversarial examples generated by our attack are hard to recognize for humans. Experiments also show that adversarial training increases model robustness against our attack. Our current attack works for models with context-independent word embeddings. In the future, we would like to extend our attack to models using contextualized word embeddings, including ELMo (Peters et al., 2018), BERT (Devlin et al., 2019), etc.

References

Naveed Akhtar and Ajmal Mian. 2018. Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access*, 6:14410–14430.

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 Conference on Empirical Methods in Natural Language Processing*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. HotFlip: White-box adversarial examples for text classification. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *Proceedings of the International Conference on Learning Representations*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Jiwei Li, Xinlei Chen, Eduard H. Hovy, and Dan Jurafsky. 2016a. Visualizing and understanding neural models in NLP. In *The Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016b. Understanding neural networks through representation erasure. *arXiv preprint arXiv:1612.08220*.

Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2019. Textbugger: Generating adversarial text against real-world applications. In *Annual Network and Distributed System Security Symposium*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *Proceedings of the International Conference on Learning Representations*.

Apostolos Modas, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. 2019. Sparsefool: a few pixels make a big difference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. 2016. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Jonathan Uesato, and Pascal Frossard. 2019. Robustness via curvature regularization, and vice versa. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Anh Nguyen, Jason Yosinski, and Jeff Clune. 2015. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *Proceedings of the International Conference on Learning Representations*.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. 2019. Robustness may be at odds with accuracy. In *Proceedings of the International Conference on Learning Representations*.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing NLP. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, November.

Huangzhao Zhang, Hao Zhou, Ning Miao, and Lei Li. 2019. Generating fluent adversarial examples for natural languages. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Zhengli Zhao, Dheeru Dua, and Sameer Singh. 2018. Generating natural adversarial examples. In *Proceedings of the International Conference on Learning Representations*.