A Simple General Method for Detecting Textual Adversarial Examples

Anonymous ACL submission

Abstract

Although deep neural networks have achieved state-of-the-art performance in various machine learning and artificial intelligence tasks, adversarial examples, constructed by adding small non-random perturbations to correctly classified inputs, successfully fool highly expressive deep classifiers into incorrect predictions. Approaches to adversarial attacks in natural language tasks have boomed in the last five years using character-level, word-level, phrase-level, or sentence-level textual perturbations. While there is some work in NLP on defending against such attacks through proactive methods, like adversarial training, there is to our knowledge no effective reactive approaches to defence via detection of textual adversarial examples such as is found in the image processing literature. In this paper, we apply distance-based ensemble learning and semantic representations from different representation learning models based on our understanding of the reason for adversarial examples to fill this gap. Our technique, MultiDistance Representation Ensemble Method (MDRE), obtains state-of-the-art results on character-level, word-level, and phrase-level attacks on the IMDB dataset as well as on the later two with respect to the MultiNLI dataset. If this paper is accepted, we will publish our code.

1 Introduction

Highly expressive deep neural networks are fragile against adversarial examples, constructed by carefully designed small perturbations of normal examples, that can fool deep classifiers to make wrong predictions (Szegedy et al., 2013). Crafting adversarial examples in images involves adding small non-random perturbations to many pixels in inputs that would be correctly classified by a target model. These perturbations can force high-efficacy models into incorrect classifications and are often imperceptible to humans (Szegedy et al., 2013; Goodfellow et al., 2014; Moosavi-Dezfooli et al., 2016; Papernot et al., 2016a; Carlini and Wagner, 2017b; Chen et al., 2018). However, when adversarial examples have been studied in the context of text, to our knowledge, only Miyato et al. (2016) aligns closely with the original intuition of adversarial examples in applying perturbations to word embeddings, which are inputs of deep neural nets. Rather, most adversarial attack techniques use semantics-preserving textual changes other than embedding perturbations, at character-level, word-level, phrase-level, or sentence-level (Pruthi et al., 2019; Jia and Liang, 2017; Alzantot et al., 2018; Ribeiro et al., 2018; Ren et al., 2019; Iyyer et al., 2018); see Table 1. This variety increases the difficulty of detecting textual adversarial examples.

Generating adversarial examples to attack deep neural nets and protecting deep neural nets from adversarial examples have been extensively studied in image classification tasks (Szegedy et al., 2013; Goodfellow et al., 2014; Moosavi-Dezfooli et al., 2016; Papernot et al., 2016a; Carlini and Wagner, 2017b; Chen et al., 2018; Papernot et al., 2016b; Feinman et al., 2017; Ma et al., 2018; Lee et al., 2018). However, in the natural language domain, only crafting of adversarial examples has been comprehensively considered (Jia and Liang, 2017; Alzantot et al., 2018; Ribeiro et al., 2018; Ren et al., 2019; Iyyer et al., 2018). Defence against textual adversaries, primarily through increasing the robustness of deep neural networks, is much less studied (Jia et al., 2019; Pruthi et al., 2019). In the image processing space, Cohen et al. (2020) refers to these as *proactive* defence methods, and Carlini and Wagner (2017a) notes that they can be evaded by optimization-based attacks, such as constructing new loss functions; in the NLP space, Yoo and Qi (2021) observes that generating word-level textual adversaries for proactive adversarial training are computationally expensive because of necessary search and constraints based on sentence encoding. Consequently, Feinman et al. (2017); Ma et al.

		Prediction
Original	**This** is a **story** of two **misfits who** don't **stand** a **chance alone**, but **together** they are **magnificent**.	Positive
Character-level (Pruthi et al., 2019)	**TZyTis** is a **sotry** of two **misifts** who don't **stad** a **ccange alUone**, but **tpgthr** they are **mgnificent**.	Negative
Word-level (Alzantot et al., 2018)	This is a **conte** of two **who** don't **stands** a **opportunities** alone, but together they are **opulent**.	Negative
Phrase-level (Iyyer et al., 2018)	Why don't you have two misfits who don't stand a chance alone, but together they're beautiful.	Negative
Sentence-level (Jia and Liang, 2017)	This is a story of two misfits who don't stand a chance alone, but together they are magnificent. **ready south hundred at size expected worked whose turn poor.**	Negative

Table 1: Examples of textual adversarial instances on a sentiment analysis task

(2018); Lee et al. (2018); Papernot and McDaniel (2018) explore *reactive* defence methods (Cohen et al., 2020) in the image processing space: these focus on distinguishing real from adversarial examples, in order to detect them before they are passed to neural networks. These reactive defences have been explored in only a limited way in the NLP space (Mozes et al., 2021).

The contribution of this paper is to propose a simple textual adversarial reactive detector, MultiDistance Representation Ensemble Method (MDRE), based on our understanding of the reason for adversarial examples, that achieves state-of-the-art results across a range of attack methods and domains.

2 Related Work

In this section, we briefly review state-of-the-art works on attacking and defending neural networks against textual adversarial examples.

Textual Adversarial Attacks: Pruthi et al. (2019) introduced four categories of character-level perturbations: swapping, dropping, adding, and keyboard mistakes. Ebrahimi et al. (2018) explored an efficient white-box gradient-based method using the gradients of a model with respect to its one-hot input vectors, is called HotFlip. Alzantot et al. (2018) and Ren et al. (2019) proposed word-level attacks through transformations, search methods, constraints, and goal functions (Morris et al., 2020), where transformations embody a single perturbation and search methods specify how to do multiple perturbations. Ribeiro et al. (2018) presented an approach to generate model-agnostic semantically equivalent adversaries (SEAs), based on paraphrase generation techniques using translation models (Mallinson et al., 2017). Iyyer et al. (2018) proposed semantics-preserving syntactically controlled paraphrase networks (SCPNs), which takes a sentence and a target syntactic form as inputs and produces sentences whose syntax conforms to the target syntactic as candidate adversarial examples. Jia and Liang (2017) generated concatenative sentence-level adversaries by adding grammatical or ungrammatical sequences to the end of a SQuAD (Rajpurkar et al., 2016) paragraphs and leaving questions and answers unchanged.

Textual Adversarial Defences: Adversarial training (Goodfellow et al., 2014) is a commonly used defence method to augment training data with adversarial examples and their correct labels, which is effective in Ribeiro et al. (2018), Ebrahimi et al. (2018), but only has limited utility in Pruthi et al. (2019), Jia and Liang (2017). Jia et al. (2019) applies interval bound propagation (IBP) to minimize an upper bound of possible candidate sentences losses when facing word substitutions adversaries. Jones et al. (2020) introduced robust encodings (RobEn) to cluster words and typos, and produced one encoding for each cluster to harness adversarial typos. Zhou et al. (2019) proposed learning to discriminate perturbations (DISP) framework to block character-level and word-level adversarial perturbations by recognising and replacing perturbed words. Mozes et al. (2021) noticed and verified a characteristic of word-level adversaries that replacement words are less likely to occur than their substitutions, therefore, they constructed a rule-based, model-agnostic frequency-guided word substitutions (FGWS) algorithm, which is the only existing textual reactive defence method as far as we know.

3 Reason for Adversarial Examples

Adversarial examples are counter-intuitive because lots of deep neural net decisions are non-interpretable so far. In this section, we try to understand how deep feedforward nets work, then reveal the reason for both image and textual adversarial examples.

Essentially, neural nets are functions composed of affine functions with nonlinear functions and mapped from a high dimensional feature space to an l-dimensional output space, denoted by $f : \mathbb{R}^n \rightarrow \mathbb{R}^l$ (Strang, 2019). Here, n represents the dimension of input feature vectors, such as image pixel value vectors or text representation vectors; l is the cardinal number of a label set $\{0, \cdots, l-1\}$ which is the number of elements in this label set.

The structure of a feedforward neural net could be expressed as follows (Strang, 2019):

$$f(\boldsymbol{v_0}) = R_s(L_s(R_{s-1}(\cdots(L_1(\boldsymbol{v_0}))))) \quad (1)$$

s means the depth of this multilayer perceptron representing the number of layers of this neural net f. $\boldsymbol{v_0} \in \mathbb{R}^n$ stands for an input feature vector from a dataset. It has n features, and those features are the n components of $\boldsymbol{v_0}$. L_i denotes an affine function, which is the linear part of the i-th layer, yielding $\boldsymbol{u_i} = L_i(\boldsymbol{v_{i-1}}) = \boldsymbol{A_i v_{i-1}} + \boldsymbol{b_i}$. The $\boldsymbol{v_{i-1}}$ is the i-th layer input vector of length N_{i-1}. The matrix $\boldsymbol{A_i}$ and the bias vector $\boldsymbol{b_i}$ are weights of the i-th layer constructed by an optimization algorithm. The output of the i-th layer is a vector $\boldsymbol{v_i} = R_i(\boldsymbol{u_i}) = R_i(L_i(\boldsymbol{v_{i-1}})) = R_i(\boldsymbol{A_i v_{i-1}} + \boldsymbol{b_i})$ of length N_i, and R_i is the nonlinear activation function of this layer, which is applied to each component of $\boldsymbol{u_i}$.

If all nonlinear activation functions in a deep feedforward network f are ReLU activation functions, Strang (2019) explains that this function f is a continuous piecewise linear function, since it is a composite function which is composed of linear parts and piecewise linear parts, and both of them are continuous. More excitingly, he illustrates that the graph of this function is a surface made up of many, many flat pieces — they are planes or hyperplanes — that fit together along all the folds where a ReLU produces a change of slope. This is like a high dimensional origami with infinite flat pieces.

On the basis of these, considering that linear parts are the same in all feedforward models and nonlinear activation functions have two categories — piecewise linear functions, such as ReLU and leaky ReLU, and curved functions, like sigmoid and tanh functions — we agree with the ideas that mentioned in Hauser and Ray (2017) and Brahma et al. (2015), and assume that the graph of a deep feedforward net function is a Riemannian manifold M embedded in input Euclidean space $\mathbb{R}^n$. Since all examples, including normal and adversarial examples, are inputs of f, they lie on M.

In addition, the same predicted examples distributed within some specific areas of this Riemannian manifold, is called a decision region.

Definition 3.1 (Decision Region (Nguyen et al., 2018))**.** The decision region of a given class $0 \leq j \leq l-1$, denoted by $\boldsymbol{C_j}$, is defined as

$$\boldsymbol{C_j} = \{\boldsymbol{v_0} \in \mathbb{R}^n | f_j(\boldsymbol{v_0}) > f_k(\boldsymbol{v_0}), \forall k \neq j\}$$

$f_k(\boldsymbol{v_0})$ is the k-class predicted value of an input vector $\boldsymbol{v_0}$. The decision region $\boldsymbol{C_j}$ stands for an area containing all examples whose predicted probabilities of the class j are higher than other classes'. $\boldsymbol{C_j}$ is a Riemannian submanifold $\widetilde{M}_j$ of M which is a subset of M (Lee, 2006). Feedforward neural nets are capable of forming disconnected decision regions (Makhoul et al., 1989; Nguyen et al., 2018), therefore, $\boldsymbol{C_j}$ could be a disconnected Riemannian submanifold, which can be separated as a union of two non-empty disjoint parts.

According to dataset distributions, samples can be divided into in-distribution and out-of-distribution samples. If a test example is from the same distribution of the training set, it is an in-distribution sample, otherwise, it is an out-of-distribution sample (Hendrycks and Gimpel, 2018). Adversarial examples are out-of-distribution samples (Lee et al., 2018).

To sum up, since adversarial examples are constructed by adding imperceptible non-random perturbations to inputs of correctly classified test examples to fool highly expressive deep neural nets into incorrect classifications (Szegedy et al., 2013), the reason for both image and textual adversarial examples is that perturbations cause normal examples to transfer from one decision region, represented by a Riemannian submanifold, to another, and they are out-of-distribution samples for the dataset and for training examples from their decision regions.

4 MultiDistance Representation Ensemble Method (MDRE)

As illustrated in Section 3, an adversarial example is generated because perturbations cause a correctly predicted test input to transfer from one decision region to another, and it is an out-of-distribution sample of training examples from its decision region. Each decision region's samples are located in a Riemannian submanifold of a Riemannian manifold M of the deep neural net function (1) which are embedded in the input Euclidean space $\mathbb{R}^n$. Therefore, even though adversarial examples, and

Algorithm 1 MultiDistance Representation Ensemble Method (MDRE)

Input:

$\mathbb{D} = \{\boldsymbol{X}^{(train)}, \boldsymbol{X}^{(norm)}, \boldsymbol{X}^{(adv)}\}$: a dataset; there are k examples in $\boldsymbol{X}^{(norm)}$ and $\boldsymbol{X}^{(adv)}$

H: an array containing m representation learning models

$g : \mathbb{R}^m \rightarrow \{0, 1\}$: a multivariate binary classification model (MDRE)

$f : \mathbb{R}^n \rightarrow \mathbb{R}^l$: a deep feedforward net that is the target model for an adversarial attack

Output:

Detection accuracy of MDRE: acc

Initializing inputs and labels of g: $\boldsymbol{x} = zeros[2k, m]$, $\boldsymbol{y} = zeros[2k]$
Computing examples' predictions from f of $\mathbb{D}$: $\{\hat{\boldsymbol{y}}^{(train)}, \hat{\boldsymbol{y}}^{(norm)}, \hat{\boldsymbol{y}}^{(adv)}\}$
for $j \in \{0, \cdots, m-1\}$ **do**
    Computing examples' representations from $H[j]$ of $\mathbb{D}$: $\{\boldsymbol{V}_j^{(train)}, \boldsymbol{V}_j^{(norm)}, \boldsymbol{V}_j^{(adv)}\}$
    for $i \in \{0, \cdots, k-1\}$ **do**
        Calculating $d_j^{(norm)}, d_j^{(adv)}$ for examples $\boldsymbol{X}_i^{(norm)}, \boldsymbol{X}_i^{(adv)}$
        $\boldsymbol{x}[i, j] = d_j^{(norm)}$, $\boldsymbol{y}[i] = 0$
        $\boldsymbol{x}[k+i, j] = d_j^{(adv)}$, $\boldsymbol{y}[k+i] = 1$
    end for
end for
Training g by randomly choosing 80% of $\{(\boldsymbol{x}_{i,:}, \boldsymbol{y}_i)\}_{i=0}^{2k-1}$
$acc =$ test accuracy of g using the rest 20% of $\{(\boldsymbol{x}_{i,:}, \boldsymbol{y}_i)\}_{i=0}^{2k-1}$

same predicted normal test and training examples lie on a same Riemannian submanifold but from different distributions.

There are various techniques to measure the difference between two distributions, such as Kullback-Leibler divergence or Wasserstein distance. The Wasserstein distance is a distance between two probability distributions on a given metric space, and can be viewed as the least accumulated moving distance to move a unit of one distribution's samples to a unit of another distribution's samples, which is assumed to be the amount of samples that need to be moved times the mean distance they have to be moved. As discussed in Section 3, since the graph of a deep feedforward net function is a Riemannian manifold, the metric should be Riemannian metrics, and we'd better use Riemannian geodesics, which are the generalizations of straight line in manifolds (Lee, 2006), to measure distances between samples. Motivated by Tenenbaum et al. (2000)'s argument that for neighboring points, an input space distance provides a good approximation to a geodesic distance, to simplify we assume that a Euclidean distance between an adversarial example a' and a''s nearest neighbor among training examples from a''s decision region is bigger than a Euclidean distance between its corresponding original normal test example a and a's nearest neighbor among training examples from a's decision region.

In natural language processing, most inputs of deep neural networks are learned representations by representation learning models nowadays. Even though current methods of representation learning are effective in various tasks (Devlin et al., 2019; Liu et al., 2019; Yang et al., 2019; Lewis et al., 2020), semantic meanings and semantic differences between texts from humans' perspective are not perfectly captured by textual representation vectors (Liu et al., 2020). In addition, as mentioned in Section 1, most textual adversarial generation algorithms do not modify representations, which are input feature vectors, but modify original texts. Therefore, the assumed characteristic of adversaries in the last paragraph may lose efficiency in language adversarial detection scenarios. To build a stronger reactive classifier, we use ensemble learning to combine distances between representations learned from multiple representation learning models. We construct a more effective MultiDistance Representation Ensemble Method (MDRE), as illustrated in Algorithm 1.

The MDRE is a multivariate supervised binary classification model $g : \mathbb{R}^m \rightarrow \{0, 1\}$. m is the number of representation learning models; g can be any multivariate binary classification model, such

as multivariate logistic regressions or deep neural nets; $\{0, 1\}$ is the output label set, with 1 corresponding to adversarial examples, 0 to normal examples.

The input of MDRE is a matrix $\boldsymbol{x}$ and each row vector of $\boldsymbol{x}$ is $\boldsymbol{x}_{i,:} = (d_0, d_1 \cdots, d_{m-1}) \in \mathbb{R}^m$. The element of this vector $d_j, 0 \leq j \leq m-1$ is a Euclidean distance between a semantic representation of a normal or adversarial example $\boldsymbol{v}$ and a representation of its nearest neighbour among the training examples from the decision region and located in the same Riemannian submanifold as $\boldsymbol{v}$ through the j-th representation learning model $H[j]$. To find a nearest neighbour, we compare Euclidean distances between $\boldsymbol{v}$ and all representations among the training examples from the decision region as $\boldsymbol{v}$ through $H[j]$. In Algorithm 1, $\boldsymbol{X}^{(norm)}$ consists of normal test examples corresponding to the elements of $\boldsymbol{X}^{(adv)}$, where the elements of $\boldsymbol{X}^{(norm)}$ have correct predictions from the target model f, but $\boldsymbol{X}^{(adv)}$ have incorrect predictions from f. The training and testing process of MDRE is same as the process of the selected model g.

5 Evaluation

In this section, we evaluate the utility of MDRE by using character-level, word-level, and phrase-level upstream attacks on sentiment analysis and natural language inference tasks, and comparing against several baselines: a language model, DISP (Zhou et al., 2019), and FGWS (Mozes et al., 2021). The experimental results demonstrate that MDRE outperforms these methods on sentiment analysis and natural language inference tasks for word-level and phrase-level attacks.

5.1 Experimental Setup

5.1.1 Tasks

We apply our approach and baselines to sentiment analysis and natural language inference tasks. The sentiment analysis task has been the most widely used testbed for generating textual adversarial examples (Pruthi et al., 2019; Alzantot et al., 2018; Ribeiro et al., 2018; Ren et al., 2019; Iyyer et al., 2018), making this the natural domain for these experiments; adversarial example generation methods have also been applied the natural language inference task (Alzantot et al., 2018; Iyyer et al., 2018), so we choose this to explore the generality of our method.

We use the IMDB dataset (Maas et al., 2011) in the sentiment analysis task, which contains 50,000 movie reviews, divided into 25,000 training examples and 25,000 test examples, labelled for positive or negative sentiment. The average number of words per review in the IMDB dataset is 262 when using the Natural Language Toolkit (NLTK) (Bird et al., 2009) to tokenize examples. To capture more semantic information from each instance, we set a maximum sequence length of the IMDB dataset to 512 for all following models.

To test the robustness of MDRE, the Multi-Genre NLI (MultiNLI) corpus (Williams et al., 2018) and its mismatched test examples, which are derived from sources that differ from the training examples, are used in the natural language inference task. The MultiNLI dataset includes 392,702 training examples and 10,000 mismatched testing examples with three classes: entailment, neutral, and contradiction. The average and maximum word numbers of the MultiNLI dataset are 34 and 416 respectively, using NLTK word tokenizer. We set the maximum sequence length for this dataset to 256.

5.1.2 Attack Methods

We implement three attack methods using character-level, word-level, and phrase-level perturbations to construct adversarial examples. For all types of attacks, we take the $\text{BERT}_{\text{BASE}}$ model (Turc et al., 2019) as the target model, indicating that adversaries have different predictions with their originals by the $\text{BERT}_{\text{BASE}}$ model.

Character-level. The character-level attack is from Pruthi et al. (2019), which applies swapping, dropping, adding, and keyboard mistakes to a randomly selected word of an original example.

- Swapping: swapping two adjacent internal characters.
- Dropping: removing an internal character.
- Adding: internally inserting a new character.
- Keyboard mistakes: substituting an internal character with one of its adjacent characters in keyboards.

Here, we set maximum numbers of perturbations to half of the maximum sequence lengths of datasets; consequently, for the IMDB dataset, the maximum number of attacks is 256, and for the MultiNLI dataset is 128. If after achieving this number, the prediction of the perturbed text is still consistent with the original example, these attacks fail, and no character-level adversarial example constructed for this original example.

Dataset	Training.	Validation.	Testing.	Correctly Predicted Test Examples	Adversarial Examples		
					character-level	word-level	phrase-level
IMDB	20,000	5,000	25,000	23,121	12,267	10,343	7,048
MultiNLI	314,162	78,540	10,000	8,070	7,159	3,047	4,230

Table 2: The number of examples used in experiments

Word-level. We use a method from Alzantot et al. (2018), which is an effective and widely cited word-level threat method. Their approach randomly selects a word in a sentence, replaces it with its synonymous and context fitted word according to the GloVe word vectors (Pennington et al., 2014), counter-fitting word vectors (Mrkšić et al., 2016), and the Google 1 billion words language model (Chelba et al., 2013), and applies population-based genetic algorithms from the natural selection using a combination of crossover and mutation to generate next adversarial generations.

While effective, the initial algorithm is somewhat inefficient and computationally expensive. In implementing this method, Jia et al. (2019) found that computing scores from the Google 1 billion words language model (Chelba et al., 2013) for each iteration in this approach causes its inefficiency; to improve this, they used a faster language model and prevented semantic drift, which is synonyms picked from previous iterations also apply the language model to select words from their neighbour lists. In our experiments, we adopt these modifications by using a faster Transformer-XL architecture (Dai et al., 2019) pretrained on the WikiText-103 dataset (Merity et al., 2016), and not allowing the semantic drift, so that we compute all test examples words' neighbours before attacks.

In this attack, we also set maximum numbers of perturbations, which are one fifth of the maximum sequence lengths; therefore, for the IMDB dataset is 102, and for the MultiNLI dataset is 51. For an original test example, if the number of attacks reaches this threshold but predictions do not change, no corresponding adversarial example is constructed for this original example.

Phrase-level. The phrase-level attack is from Ribeiro et al. (2018), which uses translators and back translators to generate adversarial examples. As far as we know, this is the only phrase-level perturbation technique that can be used for paragraph-length text. Their approach — termed semantically equivalent adversaries (SEAs) — translates an original sentences into multiple pivot languages, then translates them back to the source language. If there is a back translated sentences that is semantically equivalent to the original sentences, measured by a semantic score greater than a threshold, and it has a different prediction with the original sentences, then it is an adversarial example. Otherwise, this original example has no relevant adversaries.

The $\text{BERT}_{\text{BASE}}$ model is implemented as a target model for these three attacks, by which adversarial examples are misclassified. We apportion training sets on both datasets into training subsets and validation subsets, with an 80-20 split. After training, the models achieve 92.48% test accuracy on the IMDB dataset, and for the MultiNLI mismatched test set is 80.7%. The correctly predicted test examples are preserved for subsequent attack processes. After attacks, adversarial examples and their corresponding normal test examples maintain for following detectors as negative and positive examples. The number of examples used on IMDB and MultiNLI datasets and number of adversaries after attacks are shown in Table 2.

5.1.3 Detection Methods

We evaluate three baselines in addition to our MDRE in these experiments.

A language model. The first baseline is built from a language model since even though most attack algorithms intend to construct semantically and syntactically similar adversaries, many textual adversaries are abnormal and ungrammatical, as shown in Table 1. We use the Transformer-XL model (Dai et al., 2019) pretrained on the WikiText-103 dataset (Merity et al., 2016) from Hugging Face transformers (Wolf et al., 2020), and obtain language model scores for texts as the product of words prediction proportion scores. We construct a detection classifier by using a logistic regression model with language model scores as inputs; the model acts to learn a threshold on scores to distinguish adversarial examples. To train this detector, 80% scores are used for training and 20% for testing.

Learning to Discriminate Perturbations (DISP) (Zhou et al., 2019). Our second baseline is the DISP framework, which is the only compa-

rable technique for detecting textual adversarial examples across character-level and word-level attacks to our knowledge. DISP consists of three components: perturbation discriminator, embedding estimator, and hierarchical navigable small word graphs. The perturbation discriminator identifies a set of character-level or word-level perturbed tokens; the embedding estimator predicts embeddings for each perturbed token; then, hierarchical navigable small word graphs map these embeddings to actual words to correct adversarial perturbations. DISP is not itself designed as a adversarial example detector, but we adapt it for that task: if an adversarial example rectified by DISP predicts the same class as the target model predicts for the corresponding initial original example, or the prediction of a normal (non-adversarial) example rectified by DISP isn't changed, we consider DISP to have been successful in its detection. Otherwise, it is not. Since DISP is designed for character-level and word-level attacks, we do not consider using it for phrase-level attacks.

Frequency-guided word substitutions (FGWS) (Mozes et al., 2021). Our third baseline is FGWS. Mozes et al. (2021) noticed, and verified using hypothesis testing, that a characteristic of word-level adversaries was that replacement words are less likely to occur than their substitutions. They use this feature to construct a rule-based, model-agnostic frequency-guided word substitutions (FGWS) algorithm which distinguishes adversarial examples by replacing infrequent words in examples with their higher frequency synonyms. If the replacements cause prediction confidence changes exceeding a threshold, these examples are deemed adversarial examples.

They use WordNet (Fellbaum, 2005) and GloVe vectors (Pennington et al., 2014) to find neighbors of a word. A word frequency is its number of occurrences in the corresponding dataset's training examples; infrequent words are defined as those words whose frequencies are lower than a threshold. They set this threshold to be the frequency of the word at the $\{0\text{-th}, 10\text{-th}, \cdots, 100\text{-th}\}$ percentile of word frequencies in training set. If the prediction confidence differences between sequences with replaced words and their corresponding original sequences are higher than a threshold, the original sequences are assumed to be adversarial examples. They set this threshold to the 90%-th confidence difference between words substituted validation set and original validation set in their experiment.

We use same methods to construct thresholds and select best prediction accuracy among different frequency thresholds as FGWS's detection accuracy. We use the $\text{BERT}_{\text{BASE}}$ model to generate all predictions for input texts. FGWS is only designed to be applied to word-level attacks.

MultiDistance Representation Ensemble Method (MDRE). The key ideas behind MDRE is that (1) adversarial examples are out-of-distribution samples relative to training examples from their decision regions and (2) ensemble learning can help identify this. In order to explore the effects of these two components, we apply a MDRE_{base} model, where $m = 1$ and $H = [\text{BERT}_{\text{BASE}}]$. In MDRE, we set $m = 4$, $H = [\text{BERT}_{\text{BASE}}$, $\text{RoBERTa}_{\text{BASE}}$, $\text{XLNet}_{\text{BASE}}$, $\text{BART}_{\text{BASE}}$]. For both MDRE_{base} and MDRE, g is a logistic regression model. See Algorithm 1 for more information of notations.

5.2 Experimental Results

As shown in Table 3, the performance of the language model is similar to random guess, since the ratio between positive (normal) and negative (adversarial) examples is 1:1. We observed that language model prediction proportion scores are sensitive to the number of words in examples because each word scores is between 0 to 1 and more words leads to lower scores. In addition, in some contexts, scores for synonyms, or typos which are out-of-dictionary words, are lower but close to scores of original words, which do not have the large differences that might be expected.

DISP effectively applies the bidirectional language model feature of the BERT model and builds a powerful perturbation discriminator, which labels character-level or word-level perturbed tokens to 1, and unperturbed tokens to 0. The perturbation discriminator achieves F_1 scores of 95.06% on IMDB dataset and 97.67% on MultiNLI dataset, using their own adversaral attack methods. However, the embedding estimator predicts embeddings through inputting 5-grams with masked middle tokens to a $\text{BERT}_{\text{BASE}}$ model with one layer feed-forward head on top and outputting embeddings of these masked tokens from 300-dimensional pretrained FastText English word vectors (Mikolov et al., 2018). This is challenging and restricts the overall performance of DISP.

Intuitively, adversaries' predictions are different

Dataset	Detecting Method	Character-level Attack	Word-level Attack	Phrase-level Attack
IMDB	Language Model	0.4952	0.4966	0.4988
	DISP	0.8936	0.7714	—
	FGWS	—	0.5230	—
	MDRE$_{base}$	0.9126	0.8062	0.8904
	MDRE	**0.9236**	**0.8132**	**0.9585**
MultiNLI	Language Model	0.5021	0.4807	0.4917
	DISP	**0.7496**	0.6137	—
	FGWS	—	0.5203	—
	MDRE$_{base}$	0.6781	0.6103	0.6147
	MDRE	0.7238	**0.6423**	**0.7027**

Table 3: The accuracy for detection classifiers

from their original counterparts, which are ordinary language; therefore, adversaries may contain rare and infrequent words. According to the English word frequency dataset [1], some words frequencies in examples of Alzantot et al. (2018) are shown in Table 4. We can find that the intuition is correct

org.	org. freq.	sub.	sub. freq.
terrible	8,610,277	horrific	1,017,211
		horrifying	491,916
considered	57,378,298	regarded	6,892,622
kids	96,602,880	youngstars	—
runner	7,381,022	racer	3,625,077
battling	1,340,424	—	—
strives	1,415,683	—	—

Table 4: Original and modified sample words frequencies in examples of Alzantot et al. (2018)

that replacement words frequencies drop compared with substitutions; however, they may be higher than other normal words. Therefore, using one threshold makes it difficult to separate adversarially substituted words from all normal words. Alternative approaches to applying the characteristic of adversarial words frequencies may work better.

MDRE$_{base}$ works in detecting adversarial examples: the detection accuracy on both IMDB dataset and MultiNLI dataset, and all upstream adversarial attacks is substantially higher than random guess, and better than the baselines, except for DISP against character-level attacks on MultiNLI dataset, where MDRE is a fairly close second. The detection accuracy on MultiNLI dataset is lower than IMDB dataset, although this is not a surprise. It uses the mismatched test set of MultiNLI dataset which makes the task more challenging. The results show that MDRE is sensitive to sample distributions, so if some normal test examples are from a different distribution of training samples, such as noise examples, they will influence the performance of MDRE. Ensemble learning helps to build a stronger detector.

[1]The english word frequency: https://www.kaggle.com/rtatman/english-word-frequency

6 Conclusion and Future work

In this paper, we proposed a simple and general textual adversarial reactive detector, MultiDistance Representation Ensemble Method (MDRE), based on our understanding of the reason for adversarial examples, that they are generated because perturbations cause normal test inputs to transfer from one decision region to another, and they are out-of-distribution samples. Each decision region's samples are located in a Riemannian submanifold of a Riemannian manifold of a deep feedforward network function (1). The experimental results show MDRE achieves state-of-the-art results on detecting character-level, word-level and phrase-level adversaries on the IMDB dataset as well as on the latter two with respect to the MultiNLI dataset.

However, as discussed in Section 4, for simplicity we only implement Euclidean distances between example representations and representations of their nearest neighbors among the training examples from the same decision regions, to characterise distribution differences between adversarial examples and normal examples. Applying more probability distribution theories, as Feinman et al. (2017); Lee et al. (2018); Ma et al. (2018) did in the image processing space, may help to build better detectors. Further, we hope reactive adversarial detectors will not be restricted to feedforward deep target models, but expand to all kinds of deep neural nets which are vulnerable to adversarial attacks.

References

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896, Brussels, Belgium. Association for Computational Linguistics.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Pratik Prabhanjan Brahma, Dapeng Wu, and Yiyuan She. 2015. Why deep learning works: A manifold disentanglement perspective. *IEEE transactions on neural networks and learning systems*, 27(10):1997–2008.

Nicholas Carlini and David Wagner. 2017a. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, pages 3–14.

Nicholas Carlini and David Wagner. 2017b. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2013. One billion word benchmark for measuring progress in statistical language modeling. *arXiv preprint arXiv:1312.3005*.

Pin-Yu Chen, Yash Sharma, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. 2018. Ead: elastic-net attacks to deep neural networks via adversarial examples. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Gilad Cohen, Guillermo Sapiro, and Raja Giryes. 2020. Detecting adversarial samples using influence functions and nearest neighbors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14453–14462.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy. Association for Computational Linguistics.

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 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 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)*, pages 31–36, Melbourne, Australia. Association for Computational Linguistics.

Reuben Feinman, Ryan R Curtin, Saurabh Shintre, and Andrew B Gardner. 2017. Detecting adversarial samples from artifacts. *arXiv preprint arXiv:1703.00410*.

Christiane Fellbaum. 2005. Wordnet and wordnets. In Alex Barber, editor, *Encyclopedia of Language and Linguistics*, pages 2–665. Elsevier.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Michael Hauser and Asok Ray. 2017. Principles of riemannian geometry in neural networks. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Dan Hendrycks and Kevin Gimpel. 2018. A baseline for detecting misclassified and out-of-distribution examples in neural networks.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1875–1885, New Orleans, Louisiana. Association for Computational Linguistics.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. 2019. Certified robustness to adversarial word substitutions. 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)*, pages 4129–4142, Hong Kong, China. Association for Computational Linguistics.

Erik Jones, Robin Jia, Aditi Raghunathan, and Percy Liang. 2020. Robust encodings: A framework for combating adversarial typos. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2752–2765, Online. Association for Computational Linguistics.

John M Lee. 2006. *Riemannian manifolds: an introduction to curvature*, volume 176. Springer Science & Business Media.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. 2018. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *arXiv preprint arXiv:1807.03888*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Zhiyuan Liu, Yankai Lin, and Maosong Sun. 2020. *Representation learning for natural language processing*. Springer Nature.

Xingjun Ma, Bo Li, Yisen Wang, Sarah M Erfani, Sudanthi Wijewickrema, Grant Schoenebeck, Dawn Song, Michael E Houle, and James Bailey. 2018. Characterizing adversarial subspaces using local intrinsic dimensionality. *arXiv preprint arXiv:1801.02613*.

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 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

J. Makhoul, R. Schwartz, and A. El-Jaroudi. 1989. Classification capabilities of two-layer neural nets. In *International Conference on Acoustics, Speech, and Signal Processing,*, pages 635–638 vol.1.

Jonathan Mallinson, Rico Sennrich, and Mirella Lapata. 2017. Paraphrasing revisited with neural machine translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 881–893, Valencia, Spain. Association for Computational Linguistics.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 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).

Takeru Miyato, Andrew M Dai, and Ian Goodfellow. 2016. Adversarial training methods for semi-supervised text classification. *arXiv preprint arXiv:1605.07725*.

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*, pages 2574–2582.

John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. TextAttack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126, Online. Association for Computational Linguistics.

Maximilian Mozes, Pontus Stenetorp, Bennett Kleinberg, and Lewis Griffin. 2021. Frequency-guided word substitutions for detecting textual adversarial examples. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 171–186, Online. Association for Computational Linguistics.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting word vectors to linguistic constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 142–148, San Diego, California. Association for Computational Linguistics.

Quynh Nguyen, Mahesh Chandra Mukkamala, and Matthias Hein. 2018. Neural networks should be wide enough to learn disconnected decision regions.

Nicolas Papernot and Patrick McDaniel. 2018. Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning. *arXiv preprint arXiv:1803.04765*.

Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. 2016a. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, pages 372–387. IEEE.

Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. 2016b. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE symposium on security and privacy (SP)*, pages 582–597. IEEE.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Danish Pruthi, Bhuwan Dhingra, and Zachary C. Lipton. 2019. Combating adversarial misspellings with robust word recognition. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5582–5591, Florence, Italy. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1085–1097, Florence, Italy. Association for Computational Linguistics.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically equivalent adversarial rules for debugging NLP models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 856–865, Melbourne, Australia. Association for Computational Linguistics.

Gilbert Strang. 2019. *Linear algebra and learning from data*. Wellesley-Cambridge Press Cambridge.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Joshua B Tenenbaum, Vin De Silva, and John C Langford. 2000. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: On the importance of pre-training compact models. *arXiv preprint arXiv:1908.08962v2*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

Jin Yong Yoo and Yanjun Qi. 2021. Towards improving adversarial training of NLP models. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 945–956, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yichao Zhou, Jyun-Yu Jiang, Kai-Wei Chang, and Wei Wang. 2019. Learning to discriminate perturbations for blocking adversarial attacks in text classification. 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)*, pages 4904–4913, Hong Kong, China. Association for Computational Linguistics.