# Seq2Sick: Evaluating the Robustness of Sequence-to-Sequence Models with Adversarial Examples

**Minhao Cheng,[1] Jinfeng Yi,[2] Pin-Yu Chen,[3] Huan Zhang,[1] Cho-Jui Hsieh[1]**
[1]Department of Computer Science, UCLA, [2]JD AI Research,[3]IBM Research
{mhcheng, huanzhang, chohsieh}@cs.ucla.edu, yijinfeng@jd.com, pin-yu.chen@ibm.com

## Abstract

Crafting adversarial examples has become an important technique to evaluate the robustness of deep neural networks (DNNs). However, most existing works focus on attacking the image classification problem since its input space is continuous and output space is finite. In this paper, we study the much more challenging problem of crafting adversarial examples for sequence-to-sequence (seq2seq) models, whose inputs are discrete text strings and outputs have an almost infinite number of possibilities. To address the challenges caused by the discrete input space, we propose a projected gradient method combined with group lasso and gradient regularization. To handle the almost infinite output space, we design some novel loss functions to conduct non-overlapping attack and targeted keyword attack. We apply our algorithm to machine translation and text summarization tasks, and verify the effectiveness of the proposed algorithm: by changing less than 3 words, we can make seq2seq model to produce desired outputs with high success rates. We also use an external sentiment classifier to verify the property of preserving semantic meanings for our generated adversarial examples. On the other hand, we recognize that, compared with the well-evaluated CNN-based classifiers, seq2seq models are intrinsically more robust to adversarial attacks.

## Introduction

Adversarial attack on deep neural networks (DNNs) aims to slightly modify the inputs of DNNs and mislead them to make wrong predictions (Szegedy et al. 2013; Goodfellow, Shlens, and Szegedy 2014). This task has become a common approach to evaluate the robustness of DNNs – generally speaking, the easier an adversarial example can be generated, the less robust the DNN model is. However, models designed for different tasks are not born equal: some tasks are strictly harder to attack than others. For example, attacking an image is much easier than attacking a text string, since image space is continuous and the adversary can make arbitrarily small changes to the input. Therefore, even if most of the pixels of an image have been modified, the perturbations can still be imperceptible to humans when the accumulated distortion is small. In contrast, text strings live in a discrete space, and word-level manipulations may significantly change the meaning of the text. In this scenario, an adversary should change as few words as possible, and hence this limitation induces a sparse constraint on word-level changes. Likewise, attacking a classifier should also be much easier than attacking a model with sequence outputs. This is because different from the classification problem that has a finite set of discrete class labels, the output space of sequences may have an almost infinite number of possibilities. If we treat each sequence as a label, a targeted attack needs to find a specific one over an enormous number of possible labels, leading to a nearly zero volume in search space. This may explain why most existing works on adversarial attack focus on the image classification task, since its input space is continuous and its output space is finite.

In this paper, we study a harder problem of crafting adversarial examples for sequence-to-sequence (seq2seq) models (Sutskever, Vinyals, and Le 2014). This problem is challenging since it combines both aforementioned difficulties, i.e., discrete inputs and sequence outputs with an almost infinite number of possibilities. We choose this problem not only because it is challenging, but also because seq2seq models are widely used in many safety and security sensitive applications, e.g., machine translation (Bahdanau, Cho, and Bengio 2014), text summarization (Rush, Chopra, and Weston 2015), and speech recognition (Chan et al. 2016), thus measuring its robustness becomes critical. Specifically, we aim to examine the following questions in this study:

1. *Is it possible to slightly modify the inputs of seq2seq models while significantly change their outputs?*
2. *Are seq2seq models more robust than the well-evaluated CNN-based image classifiers?*

We provide an affirmative answer to the first question by developing an effective adversarial attack framework called Seq2Sick. It is an optimization-based framework that aims to learn an input sequence that is close enough to the original sequence (in terms of distance metrics in word embedding spaces or sentiment classification) while leads to the desired outputs with high confidence. To address the challenges caused by the discrete input space, we propose to use the projected gradient descent method combined with group lasso and gradient regularization. To address the challenges

of almost infinite output space, we design some novel loss functions for the tasks of non-overlapping attack and targeted keyword attack. Our experimental results show that the proposed framework yields high success rates in both tasks. However, even if the proposed approach can successfully attack seq2seq models, our answer to the second question is "Yes". Compared with CNN-based classifiers that are highly sensitive to adversarial examples, seq2seq model is intrinsically more robust since it has discrete input space and the output space is exponentially large. As a result, adversarial examples of seq2seq models usually have larger distortions and are more perceptible than the adversarial examples crafted for CNN-based image classifiers. To the best of our knowledge, this paper is the first work that evaluates the robustness of seq2seq model, which has inspired many follow-up works and has been cited since its debut.

## Related work and Background

Papernot et al.(2016) first uses Fast Gradient Sign Method (FGSM) to conduct an attack on RNN/LSTM-based classification problems. In order to generate text adversarial examples, Li, Monroe, and Jurafsky(2016) proposes to use reinforcement learning to locate important words that could be deleted in sentiment classification. Samanta and Mehta(2017) and Liang et al.(2017) generate adversarial sequences by inserting or replacing existing words with typos and synonyms. Gao et al.(2018) aims to attack sentiment classification models in a black-box setting. It develops some scoring functions to find the most important words to modify. Yang et al.(2018) applied a greedy approach and a Gumbel trick to speed up the inference time. Alzantot et al.(2018) proposed a genetic algorithm to attack sentiment analysis. These approaches differ from our method in that they study simple text classification problems while we focus on the more challenging seq2seq model with sequential outputs. Other than attacking text classifiers, Jia and Liang(2017) aims to fool reading comprehension systems by adding misleading sentences, which has a different focus than ours. Zhao, Dua, and Singh(2017) uses the generative adversarial network (GAN) to craft natural adversarial examples. However, it can only perform the untargeted attack and also suffers from high computational cost.

Notably, almost all the previous methods are based on greedy search, i.e., at each step, they search for the best word and the best position to replace the previous word. As a result, their search space grows rapidly as the length of input sequence increases. To address this issue, we propose a novel approach that uses group lasso regularization and the projected gradient descent method with gradient regularization to simultaneously search all the replacement positions. Table 1 summarizes the key differences between the proposed framework Seq2Sick and the existing attack methods on RNN-based models. Note that our paper was the first method for attacking seq2seq model on arXiv and after our work, there are some followup papers such as (Michel et al. 2019), where they use several similarity metrics to conduct the attack while our work are focusing on the BLEU score and self-defined loss functions.

Before introducing the proposed algorithms, we first briefly describe the sequence-to-sequence (seq2seq) model. Let $\mathbf{x}_i \in \mathbb{R}^d$ be the embedding vector of each input word, $N$ be the input sequence length, and $M$ be the output sequence length. Let $\omega$ be the input vocabulary, and the output word $\mathbf{y}_j \in \nu$ where $\nu$ is the output vocabulary. The seq2seq model has an encoder-decoder framework that aims at mapping an input sequence of vectors $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_N)$ to the output sequence $\mathbf{Y} = \{\mathbf{y}_1, \dots, \mathbf{y}_M\}$. Its encoder first reads the input sequence, then each RNN/LSTM cell computes $\mathbf{h}_t = f(\mathbf{x}_t,\ \mathbf{h}_{t-1})$, where $\mathbf{x}_t$ is the current input, $\mathbf{h}_{t-1}$ and $\mathbf{h}_t$ represent the previous and current cells' hidden states, respectively. The next step computes the context vector $\mathbf{c}$ using all the hidden layers of cells $\mathbf{h}_1, \dots, \mathbf{h}_N$, i.e $\mathbf{c} = q(\mathbf{h}_1, \cdots, \mathbf{h}_N)$, where $q(\cdot)$ could be a linear or non-linear function. In this paper, we follow the setting in (Sutskever, Vinyals, and Le 2014) that $\mathbf{c} = q(\mathbf{h}_1, \cdots, \mathbf{h}_N) = \mathbf{h}_N$.

Given the context vector $\mathbf{c}$ and all the previously words $\{\mathbf{y}_1, \dots, \mathbf{y}_{t-1}\}$, the decoder is trained to predict the next word $\mathbf{y}_t$. Specifically, the $t$-th cell in the decoder receives its previous cell's output $\mathbf{y}_{t-1}$ and the context vector $\mathbf{c}$, and then outputs

$$z_t = g(\mathbf{y}_{t-1}, \mathbf{c}) \text{ and } p_t = \text{softmax}(z_t), \tag{1}$$

where $g$ is another RNN/LSTM cell function. $z_t := [z_t^{(1)}, z_t^{(2)}, \dots, z_t^{(|\nu|)}] \in \mathbb{R}^{|\nu|}$ is a vector of the *logits* for each possible word in the output vocabulary $\nu$.

## Seq2Sick: Proposed Framework

Crafting adversarial examples against the seq2seq model can be formulated as an optimization problem:

$$\min_{\boldsymbol{\delta}} \ L(\mathbf{X} + \boldsymbol{\delta}) + \lambda \cdot R(\boldsymbol{\delta}), \tag{2}$$

where $R(\cdot)$ indicates the regularization function to measure the magnitude of distortions. $L(\cdot)$ is the loss function to penalize the unsuccessful attack and it may take different forms in different attack scenarios. A common choice for $R(\boldsymbol{\delta})$ is the $\ell_2$ penalty $\|\boldsymbol{\delta}\|_2^2$, but it is, as we will show later, not suitable for attacking seq2seq model. $\lambda > 0$ is the regularization parameter that balances the distortion and attack success rate – a smaller $\lambda$ will make the attack more likely to succeed but with the price of larger distortion.

In this work, we focus on two kinds of attacks: *non-overlapping attack* and *targeted keywords attack*. The first attack requires that the output of the adversarial example shares no overlapping words with the original output. This task is strictly harder than untargeted attack, which only requires that the adversarial output to be different from the original output (Zhao, Dua, and Singh 2017; Ebrahimi et al. 2017). We ignore the task of untargeted attack since it is trivial for the proposed framework, which can easily achieve a 100% attack success rate, while Ebrahimi et al.(2017) could achieve 76.24% attack success rate for text summarization and 98.8% success rate for machine translation with 1 word change. Targeted keywords attack is an even more challenging task than non-overlapping attack. Given a set of targeted keywords, the goal of targeted keywords attack is to find an adversarial input sequence such that all the keywords must appear in its corresponding output. In the following, we respectively introduce the loss functions developed for the two attack approaches.

Table 1: Summary of existing works that are designed to attack RNN models. "BINARY" indicates the attack is for binary classifications, and there is no difference between untargeted and targeted attack in this case. "CLASS" means targeted attack to a specific class. "KEYWORD" means targeted attack to a specific keyword. Here we omit follow-up works based on Seq2Sick.

| Methods | Gradient Based? | Word-level RNN? | Sequential Output? | Targeted Attack? |
|---|---|---|---|---|
| Ebrahimi et al.(2017) | $\surd$ | $\times$ | $\surd$ | Class |
| Jia and Liang(2017) | $\times$ | $\surd$ | $\times$ | $\times$ |
| Li, Monroe, and Jurafsky(2016) | $\surd$ | $\surd$ | $\times$ | Class |
| Papernot et al.(2016) | $\surd$ | $\times$ | $\surd$ | $\times$ |
| Gao et al.(2018) | $\times$ | $\surd$ | $\times$ | Binary |
| Samanta and Mehta(2017) | $\times$ | $\times$ | $\times$ | Binary |
| Zhao, Dua, and Singh(2017) | / | $\surd$ | $\surd$ | Class |
| Liang et al.(2017) | $\surd$ | $\times$ | $\times$ | Class |
| Alzantot et al.(2018) | $\times$ | $\surd$ | $\times$ | Class |
| Yang et al.(2018) | $\times$ | $\surd$ | $\times$ | Class |
| Seq2Sick (Ours) | $\surd$ | $\surd$ | $\surd$ | Keyword |

**Non-overlapping Attack** To formally define the non-overlapping attack, we let $\mathbf{s} = \{s_1, \ldots, s_M\}$ be the original output sequence, where $s_i$ denotes the location of the $i$-th word in the output vocabulary $\nu$. $\{z_1, \ldots, z_M\}$ indicates the logit layer outputs of the adversarial example. In the non-overlapping attack, the output of adversarial example should be entirely different from the original output $\mathbf{S}$, i.e.,

$$s_t \neq \arg\max_{y \in \nu} z_t^{(y)}, \ \ \forall t = 1, \ldots, M,$$

which is equivalent to

$$z_t^{(s_t)} < \max_{y \in \nu,\, y \neq s_t} z_t^{(y)}, \ \ \forall t = 1, \ldots, M.$$

Given this observation, we can define a hinge-like loss function $L$ to generate adversarial examples in the non-overlapping attack, i.e.,

$$L_{\text{non-overlapping}} = \sum\nolimits_{t=1}^{M} \max\{-\epsilon,\, z_t^{(s_t)} - \max_{y \neq s_t}\{z_t^{(y)}\}\}, \quad (3)$$

where $\epsilon \geq 0$ denotes the confidence margin parameter. Generally speaking, a larger $\epsilon$ will lead to a more confident output and a higher success rate, but with the cost of more iterations and longer running time.

We note that non-overlapping attack is much more challenging than untargeted attack, which suffices to find a one-word difference from the original output (Zhao, Dua, and Singh 2017; Ebrahimi et al. 2017). We do not take untargeted attack into account since it is straightforward and the replaced words could be some less important words such as "the" and "a".

**Targeted Keywords Attack** Given a set of targeted keywords, the goal of targeted keywords attack is to generate an adversarial input sequence to ensure that all the targeted keywords appear in the output sequence. This task is important since it suggests adding a few malicious keywords can completely change the meaning of the output sequence. For example, in English to German translation, an input sentence "*policeman helps protesters to keep the assembly in order*" should generate an output sentence "*Polizist hilft Demonstranten, die Versammlung in Ordnung zu halten*". However, changing only one word from "hilft" to "verhaftet" in the output will significantly change its meaning, as the new sentence means "*police officer arrested protesters to keep the assembly in order*".

In our method, we do not specify the positions of the targeted keywords in the output sentence. Instead, it is more natural to design a loss function that allows the targeted keywords to become the top-1 prediction at any positions. The attack is considered as successful only when ALL the targeted keywords appear in the output sequence. Therefore, the more targeted keywords there are, the harder the attack is. To illustrate our method, we start from the simpler case with only one targeted keyword $k_1$. To ensure that the target keyword word's logit $z_t^{(k_1)}$ be the largest among all the words at a position $t$, we design the following loss function:

$$L = \min_{t \in [M]} \{\max\{-\epsilon,\, \max_{y \neq k_1}\{z_t^{(y)}\} - z_t^{(k_1)}\}\}, \quad (4)$$

which essentially searches the minimum of the hinge-like loss terms over all the possible locations $t \in [M]$. When there exist more than one targeted keywords $K = \{k_1, k_2, \ldots, k_{|K|}\}$, where $k_i$ denotes the $i$-th word in output vocabulary $\nu$, we follow the same idea to define the loss function as follows:

$$L_{\text{keywords}} = \sum_{i=1}^{|K|} \min_{t \in [M]} \{\max\{-\epsilon, \max_{y \neq k_i}\{z_t^{(y)}\} - z_t^{(k_i)}\}\}. \quad (5)$$

However, the loss defined in (5) suffers from the "keyword collision" problem. When there are more than one keyword, it is possible that multiple keywords compete at the same position to attack. To address this issue, we define a mask function $m$ to mask off the position if it has been already occupied by one of the targeted keywords:

$$m_t(x) = \begin{cases} +\infty & \text{if } \arg\max_{i \in \nu} z_t^{(i)} \in K \\ x & \text{otherwise} \end{cases} \quad (6)$$

In other words, if any of the keywords appear at position $t$ as the top-1 word, we ignore that position and only consider

other positions for the placement of remaining keywords. By incorporating the mask function, the final loss for targeted keyword attack becomes:

$$\sum_{i=1}^{|K|} \min_{t\in[M]} \{m_t(\max\{-\epsilon,\ \max_{y\neq k_i}\{z_t^{(y)}\} - z_t^{(k_i)}\})\}. \quad (7)$$

## Handling Discrete Input Space

As mentioned before, the problem of "discrete input space" is one of the major challenges in attacking seq2seq model. Let $\mathbb{W}$ be the set of word embeddings of all words in the input vocabulary. A naive approach is to first learn $\mathbf{X}+\boldsymbol{\delta}^*$ in the continuous space by solving the problem (2), and then search for its nearest word embedding in $\mathbb{W}$. This idea has been used in attacking sequence classification models in Gong et al.(2018). Unfortunately, when applying this idea to targeted keywords attack, we report that all of the 100 attacked sequences on Gigaword dataset failed to generate the targeted keywords. The main reason is that by directly solving (2), the final solution will not be a feasible word embedding in $\mathbb{W}$, and its nearest neighbor could be far away from it due to the curse of dimensionality (Friedman 1997).

To address this issue, we propose to add an additional constraint to enforce that $\mathbf{X}+\boldsymbol{\delta}$ belongs to the input vocabulary $\mathbb{W}$. The optimization problem then becomes

$$\begin{aligned} \min_{\boldsymbol{\delta}} \quad & L(\mathbf{X}+\boldsymbol{\delta}) + \lambda \cdot R(\boldsymbol{\delta}) \\ \text{s.t.} \quad & \mathbf{x}_i + \delta_i \in \mathbb{W} \quad \forall i = 1,\dots,N \end{aligned} \quad (8)$$

We then apply projected gradient descent to solve this constrained problem. At each iteration, we project the current solution $\mathbf{x}_i+\delta_i$, where $\delta_i$ denotes the $i$-th column of $\boldsymbol{\delta}$, back into $\mathbb{W}$ to ensure that $\mathbf{X}+\boldsymbol{\delta}$ can map to a specific input word.

**Group lasso Regularization:** $\ell_2$ norm has been widely used in the adversarial machine learning literature to measure distortions. However, it is not suitable for our task since almost all the learned $\{\delta_t\}_{t=1}^M$ using $\ell_2$ regularization will be nonzero. As a result, most of the inputs words will be perturbed to another word, leading to an adversarial sequence that is significantly different from the input sequence.

To solve this problem, we treat each $\delta_t$ with $d$ variables as a group, and use the group lasso regularization

$$R(\boldsymbol{\delta}) = \sum\nolimits_{t=1}^{N} \|\delta_t\|_2$$

to enforce the group sparsity: only a few groups (words) in the optimal solution $\boldsymbol{\delta}^*$ are allowed to be nonzero.

## Gradient Regularization

When attacking the seq2seq model, it is common to find that the adversarial example is located in a region with very few or even no embedding vector. This will negatively affect our projected gradient method since even the closest embedding from those regions can be far away.

To address this issue, we propose a gradient regularization to make $\mathbf{X}+\boldsymbol{\delta}$ close to the word embedding space. Our final objective function becomes:

**Algorithm 1** Seq2Sick algorithm

**Input:** input sequence $\mathbf{x} = \{x_1,\dots,x_N\}$, seq2seq model, target keyword $\{k_1,\dots,k_T\}$
**Output:** adversarial sequence $\mathbf{x}^* = \mathbf{x}+\boldsymbol{\delta}^*$
Let $\mathbf{s} = \{s_1,\dots,s_M\}$ denote the original output of $\mathbf{x}$.
Set the loss $L(\cdot)$ in (9) to be (3)
**if** Targeted Keyword Attack **then**
  Set the loss $L(\cdot)$ in (9) to be (7)
**end if**
**for** $r = 1,2,\dots,T$ **do**
  back-propagation $L$ to achieve gradient $\nabla_{\boldsymbol{\delta}} L(\mathbf{x}+\boldsymbol{\delta}_r)$
  **for** $i = 1,2,\dots,N$ **do**
    $\delta_{r,i} = 0$
    **if** $\|\delta_{r,i}\| > \eta\lambda_1$ **then**
      $\delta_{r,i} = \delta_{r,i} - \eta\lambda_1 \frac{\delta_{r,i}}{\|\delta_{r,i}\|}$
    **end if**
  **end for**
  $y^{r+1} = \boldsymbol{\delta}^r + \eta \cdot \nabla_{\boldsymbol{\delta}} L(\mathbf{x}+\boldsymbol{\delta}^r)$
  $\boldsymbol{\delta}^{r+1} = \underset{\mathbf{x}+\boldsymbol{\delta}^{r+1}\in\mathbb{W}}{\arg\min} \left\| y^{r+1} - \boldsymbol{\delta}^{r+1} \right\|$
**end for**
$\boldsymbol{\delta}^* = \boldsymbol{\delta}^T$
$\mathbf{x}^* = \mathbf{x}+\boldsymbol{\delta}^*$
**return** $\mathbf{x}^*$

$$\begin{aligned} \min_{\boldsymbol{\delta}} & L(\mathbf{X}+\boldsymbol{\delta}) + \lambda_1 \sum_{i=1}^{N} \|\delta_i\|_2 + \lambda_2 \sum_{i=1}^{N} \min_{\mathbf{w}_j\in\mathbb{W}} \{\left\|\mathbf{x}_i + \delta_i - \mathbf{w}_j\right\|_2\} \\ \text{s.t. } & \mathbf{x}_i + \delta_i \in \mathbb{W} \quad \forall i = 1,\dots,N \end{aligned} \quad (9)$$

where the third term is our gradient regularization that penalizes a large distance to the nearest point in $\mathbb{W}$. The gradient of this term can be efficiently computed since it is only related to one $\mathbf{w}_j$ that has a minimum distance from $\mathbf{x}_i+\delta_i$. For the other terms, we use the proximal operator to optimize the group lasso regularization, and the gradient of the loss function $L$ can be computed through back-propagation. The detailed steps of our approach, Seq2Sick, is presented in Algorithm 1. Our source code is publicly available at https://github.com/cmhcbb/Seq2Sick.

**Computational Cost:** Our algorithm needs only one back-propagation to compute the gradient $\nabla_{\boldsymbol{\delta}} L(x+\boldsymbol{\delta})$. The bottleneck here is to project the solution back into the word embedding space, which depends on the number of words in the input dictionary of the model. Gong et al.(2018) uses $GloVe$ word embedding (Pennington, Socher, and Manning 2014) that contains millions of words to do a nearest neighbor search. Fortunately, our model does not need to use any pre-trained word embedding, thus making it a more generic attack that does not depend on pre-trained word embedding. Besides, we can employ approximate nearest neighbor (ANN) approaches to further speed up the projection step.

## Experiments

We conduct experiments on two widely-used applications of seq2seq model: text summarization and machine translation.

### Datasets

We use three datasets DUC2003, DUC2004, and Gigaword, to conduct our attack for the text summarization task. Among them, DUC2003 and DUC2004 are widely-used datasets in documentation summarization. We also include a subset of randomly chosen samples from Gigaword to further evaluate the performance of our algorithm. For the machine translation task, we use 500 samples from WMT'16 Multimodal Translation task. The statistics about the datasets are shown in Table 2.

Table 2: Statistics of the datasets. "# Samples" is the number of test examples we used for robustness evaluations

| DATASETS | # SAMPLES | AVERAGE INPUT LENGTHS |
|---|---|---|
| GIGAWORD | 1,000 | 30.1 WORDS |
| DUC2003 | 624 | 35.5 WORDS |
| DUC2004 | 500 | 35.6 WORDS |
| MULTI30K | 500 | 11.5 WORDS |

### Seq2seq models

We implement both text summarization and machine translation models on OpenNMT-py. Specifically, we use a word-level LSTM encoder and a word-based attention decoder for both applications (Bahdanau, Cho, and Bengio 2014). For the text summarization task, we use 380k training pairs from Gigaword dataset to train a seq2seq model. The architecture consists of a 2-layer stacked LSTM with 500 hidden units. We conduct experiments on two types of models, one uses the pre-trained 300-dimensional GloVe word embeddings and the other one is trained from scratch. We set the beam search size to be 5 as suggested. For the machine translation task, we train our model using 453k pairs from the Europal corpus of German-English WMT 15, common crawl and news-commentary. We use the hyper-parameters suggested by OpenNMT for both models, and have reproduced the performance reported in Rush, Chopra, and Weston(2015) and Ha, Niehues, and Waibel(2016).

### Empirical Results

**Text Summarization** For the non-overlapping attack, we use the proposed loss (3) in our objective function. A non-overlapping attack is treated as successful only if there is no common word at every position between output sequence and original sequence. We set $\lambda = 1$ in all non-overlapping experiments. Table 3 summarizes the experimental results. It shows that our algorithm only needs to change 2 or 3 words on average and can generate entirely different outputs for more than $80\%$ of sentences. We have also included some adversarial examples in Table 8. From these examples, we can only change one word to let output sequence look completely different with the original one and change the sentence's meaning completely.

Table 3: Results of non-overlapping attack in text summarization. **# changed** is how many words are changed in the input sentence. The high BLEU scores and low average number of changed words indicate that the crafted adversarial inputs are very similar to their originals, and we achieve high success rates to generate a summarization that differs with the original *at every position* for all three datasets.

| Dataset | Success% | BLEU | # changed |
|---|---|---|---|
| Gigaword | 86.0% | 0.828 | 2.17 |
| DUC2003 | 85.2% | 0.774 | 2.90 |
| DUC2004 | 84.2% | 0.816 | 2.50 |

For the targeted keywords attack, we randomly choose some targeted keywords from the output vocabulary after removing the stop words like "a" and "the". A targeted keywords attack is treated as successful only if the output sequence contains all the targeted keywords. We set $\lambda_1 = \lambda_2 = 1$ in our objective function (9) in all our experiments. Table 4 summarizes the performance, including the overall success rate, average BLEU score (Papineni et al. 2002), and the average number of changed words in input sentences. Average BLEU score is defined by exponential average over BLEU 1,2,3,4, which is commonly used in evaluating the quality of text which has been machine-translated from one natural language to another. Also, we have included some adversarial examples crafted by our method in Table 9. In Table 9, some adversarial examples with 3 sets of keywords, where "##" stands for a two-digit number after standard preprocessing in text summarization. Through these examples, our method could generate totally irrelevant subjects, verbs, numerals and objects which could easily be formed as a complete sentence with only several word changes. Note that there are three important techniques used in our algorithm: projected gradient method, group lasso, and gradient regularization. Therefore, we conduct experiments to verify the importance of each of these techniques.

**Machine Translation** We then conduct both non-overlapping and targeted keywords attacks to the English-German machine translation model. We first filter out stop words like "Ein"(a), "und"(and) in German vocabulary and randomly choose several nouns, verbs, adjectives or adverbs in German as targeted keywords. Similar to the text summarization experiments, we set $\lambda_1 = \lambda_2 = 1$ in our objective function. The success rates, BLEU scores, and the average number of words changed are reported in Table 5, with some adversarial examples shown in Table 7.

### Analysis of Syntactic structure and Semantic Meaning Preservation

In our algorithm we aim to make adversarial examples having similar meaning to original examples by constraining the number of changed words and enforcing the changed words are close to the original words in the embedding space. However, depending on the implemented word embedding techniques, in general there is no guarantee that every word pair

Table 4: Results of targeted keywords attack in text summarization. $|K|$ is the number of keywords. We found that our method can make the summarization include 1 or 2 target keywords with a high success rate, while the changes made to the input sentences are relatively small, as indicated by the high BLEU scores and low average number of changed words. When $|K| = 3$, this task becomes more challenging, but our algorithm can still find many adversarial examples.

| Datasest | $\|K\|$ | Success% | BLEU | # changed |
|---|---|---|---|---|
| Gigaword | 1 | 99.8% | 0.801 | 2.04 |
| | 2 | 96.5% | 0.523 | 4.96 |
| | 3 | 43.0% | 0.413 | 8.86 |
| DUC2003 | 1 | 99.6% | 0.782 | 2.25 |
| | 2 | 87.6% | 0.457 | 5.57 |
| | 3 | 38.3% | 0.376 | 9.35 |
| DUC2004 | 1 | 99.6% | 0.773 | 2.21 |
| | 2 | 87.8% | 0.421 | 5.1 |
| | 3 | 37.4% | 0.340 | 9.3 |

Table 5: Results of non-overlapping method and targeted keywords method in machine translation.

| Method | Success% | BLEU | # changed |
|---|---|---|---|
| Non-overlap | 89.4% | 0.349 | 3.5 |
| 1-keyword | 100.0% | 0.705 | 1.8 |
| 2-keyword | 91.0 % | 0.303 | 4.0 |
| 3-keyword | 69.6% | 0.205 | 5.3 |

close in the embedding space have similar meanings. Therefore, we have conducted additional experiments to verify the syntactic and semantic quality of our generated adversarial examples. For syntactic structure part, as showed in Table 6, we measure the perplexity of generated adversarial sentences in DUC2003 and DUC2004 dataset. It shows that our examples keeps the original syntactic structure. For the semantic meaning part, We use DeepAI's online sentiment analysis API to test whether our attack changes the sentiment of 500 sentences from DUC2003 dataset in summarization task. The results show that **only 2.2% of adversarial examples have semantic meaning differ from the original sentences**. It proves that almost all adversarial examples keep the same semantic classification unchanged.

Table 6: Perplexity score for adversarial example

| | DUC2003 | DUC2004 |
|---|---|---|
| Original | 102.02 | 121.09 |
| Non-overlap | 114.02 | 149.15 |
| 1-keyword | 159.54 | 199.01 |
| 2-keyword | 352.12 | 384.80 |

## Analysis and Discussions

**Observation from adversarial example** As shown in Table 9, our targeted keyword attack wouldn't just directly replace the keyword with some word in the source input. However, the word changed in the adversarial example and the target keyword are co-occurrent in the training dataset. It infers that seq2seq model learns the relationship between changed word and target keyword. However, the model fails to decide where it should focus on, which is strongly related with attention layer used in the model. It encourages us to use self-attention such as transformer (Vaswani et al. 2017) instead to extract all the attentions between any two words.When attacking subword transformer model, the target 1 keyword attack has 17% lower success rate and 0.13 lower BLEU score. It shows transformer model has a greater adversarial robustness.

**Robustness of Seq2Seq Model** Although our algorithm can achieve very good success rates ($84\% - 100\%$) in both non-overlapping and targeted keywords attacks with 1 or 2 keywords, we also recognize some strengths of the seq2seq model: (i) unlike CNN models where targeted attack can be conducted easily with almost 100% success rate and very small distortion that cannot be perceived by human eyes (Carlini and Wagner 2017), it is harder to turn the entire seq2seq output into a particular sentence – some sentences are even impossible to generate by seq2seq models; and (ii) since the input space of seq2seq is discrete, it is easier for human to detect the differences between the adversarial sequence and the original one, even if we only change one or few words. Therefore, we conclude that, compared with the DNN models designed for other tasks such as image classification, seq2seq models are more robust to adversarial attacks. The main reason, as pointed out in the introduction, is that the seq2seq model has a finite and discrete input space and almost infinite output space, so it is more robust than visual classification models that have an infinite and continuous input space and a very small output space (e.g., 10 categories in MNIST and 1,000 categories in ImageNet).

## Conclusion

In this paper, we propose a novel framework, i.e., Seq2Sick, to generate adversarial examples for sequence-to-sequence neural network models. We propose a projected gradient method to address the issue of discrete input space, adopt group lasso to enforce the sparsity of the distortion, and develop a regularization technique to further improve the success rate. Besides, different from most existing algorithms that are designed for untargeted attack and classification tasks, our algorithm can perform the more challenging targeted keywords attack. Our experimental results show that the proposed framework is powerful and effective: it can achieve high success rates in both non-overlapping and targeted keywords attacks with relatively small distortions and preserve similar sentiment classification results for the most of the generated adversarial examples.

Table 7: Machine translation adversarial examples. Upper 4 lines: non-overlap; Bottom 4 lines: targeted keyword ”Hund sitzt”

| | |
|---|---|
| Source input seq | A child is splashing in the water. |
| Adv input seq | A children is **unionists** in the water. |
| Source output seq | Ein Kind im Wasser. |
| Adv output seq | **Kinder sind in der Wasser @-@ <unk>**. |
| Source input seq | Two men wearing swim trunks jump in the air at a moderately populated beach. |
| Adv input seq | Two men wearing **dog Leon comes** in the air at a moderately populated beach. |
| Source output seq | Zwei Männer in Badehosen springen auf einem mäßig belebten Strand in die Luft. |
| Adv output seq | Zwei Männer tragen **Hund** , der in der Luft **sitzt** , hat <unk> <unk> . |

Table 8: Text summarization adversarial examples using non-overlapping method. Surprisingly, it is possible to make the output sequence completely different by changing only one word in the input sequence.

| | |
|---|---|
| Source input seq | among asia 's leaders , prime minister mahathir mohamad was notable as a man with a bold vision : a physical and social transformation that would push this nation into the forefront of world affairs . |
| Adv input seq | among **lynn** 's leaders , prime minister mahathir mohamad was notable as a man with a bold vision : a physical and social transformation that would push this nation into the forefront of world affairs. |
| Source output seq | asia 's leaders are a man of the world |
| Adv output seq | **a vision for the world** |
| Source input seq | under nato threat to end his punishing offensive against ethnic albanian separatists in kosovo , president slobodan milosevic of yugoslavia has ordered most units of his army back to their barracks and may well avoid an attack by the alliance , military observers and diplomats say |
| Adv input seq | under nato threat to end his punishing offensive against ethnic albanian separatists in kosovo , president slobodan milosevic of yugoslavia has **jean-sebastien** most units of his army back to their barracks and may well avoid an attack by the alliance , military observers and diplomats say. |
| Source output seq | milosevic orders army back to barracks |
| Adv output seq | **nato may not attack kosovo** |
| Source input seq | flooding on the yangtze river remains serious although water levels on parts of the river decreased today , according to the state headquarters of flood control and drought relief . |
| Adv input seq | flooding **that** the yangtze river **becomes** serious although water levels on parts of the river decreased today , according to the state headquarters of flood control and drought relief . |
| Source output seq | floods on yangtze river continue |
| Adv output seq | **flooding in water recedes in river** |

Table 9: Text summarization examples with targeted keywords “police arrest ”

| | |
|---|---|
| Source input seq | north korea is entering its fourth winter of chronic food shortages with its people malnourished and at risk of dying from normally curable illnesses , senior red cross officials said tuesday. |
| Adv input seq | north **detectives** is **apprehended** its fourth winter of chronic food shortages with its people malnourished and at risk of dying from normally curable illnesses , senior red cross officials said tuesday. |
| Source output seq | north korea enters fourth winter of food shortages |
| Adv output seq | north **police arrest** fourth winter of food shortages. |
| Source input seq | after a day of fighting , congolese rebels said sunday they had entered kindu , the strategic town and airbase in eastern congo used by the government to halt their advances. |
| Adv input seq | after a day of fighting , **nordic detectives** said sunday they had entered **UNK** , the strategic town and airbase in eastern congo used by the government to halt their advances. |
| Source output seq | congolese rebels say they have entered UNK. |
| Adv output seq | nordic **police arrest** ## in congo. |
| Source input seq | president boris yeltsin stayed home tuesday , nursing a respiratory infection that forced him to cut short a foreign trip and revived concerns about his ability to govern. |
| Adv input seq | president boris yeltsin stayed home tuesday , **cops cops** respiratory infection that forced him to cut short a foreign trip and revived concerns about his ability to govern. |
| Source output seq | yeltsin stays home after illness |
| Adv output seq | yeltsin stays home after **police arrest** |

## References


Alzantot, M.; Sharma, Y.; Elgohary, A.; Ho, B.-J.; Srivastava, M.; and Chang, K.-W. 2018. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2890–2896.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Carlini, N., and Wagner, D. 2017. Towards evaluating the robustness of neural networks. In *Security and Privacy (SP), 2017 IEEE Symposium on*, 39–57. IEEE.

Chan, W.; Jaitly, N.; Le, Q.; and Vinyals, O. 2016. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4960–4964. IEEE.

Ebrahimi, J.; Rao, A.; Lowd, D.; and Dou, D. 2017. Hotflip: White-box adversarial examples for nlp. *arXiv preprint arXiv:1712.06751*.

Friedman, J. H. 1997. On bias, variance, 0/1—loss, and the curse-of-dimensionality. *Data mining and knowledge discovery* 1(1):55–77.

Gao, J.; Lanchantin, J.; Soffa, M. L.; and Qi, Y. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. *arXiv preprint arXiv:1801.04354*.

Gong, Z.; Wang, W.; Li, B.; Song, D.; and Ku, W.-S. 2018. Adversarial texts with gradient methods. *arXiv preprint arXiv:1801.07175*.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Ha, T.-L.; Niehues, J.; and Waibel, A. 2016. Toward multilingual neural machine translation with universal encoder and decoder. *arXiv preprint arXiv:1611.04798*.

Jia, R., and Liang, P. 2017. Adversarial examples for evaluating reading comprehension systems. *arXiv preprint arXiv:1707.07328*.

Li, J.; Monroe, W.; and Jurafsky, D. 2016. Understanding neural networks through representation erasure. *arXiv preprint arXiv:1612.08220*.

Liang, B.; Li, H.; Su, M.; Bian, P.; Li, X.; and Shi, W. 2017. Deep text classification can be fooled. *arXiv preprint arXiv:1704.08006*.

Michel, P.; Li, X.; Neubig, G.; and Pino, J. M. 2019. On evaluation of adversarial perturbations for sequence-to-sequence models. *arXiv preprint arXiv:1903.06620*.

Papernot, N.; McDaniel, P.; Swami, A.; and Harang, R. 2016. Crafting adversarial input sequences for recurrent neural networks. In *Military Communications Conference, MILCOM 2016-2016 IEEE*, 49–54. IEEE.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, 311–318. Association for Computational Linguistics.

Pennington, J.; Socher, R.; and Manning, C. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

Rush, A. M.; Chopra, S.; and Weston, J. 2015. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*.

Samanta, S., and Mehta, S. 2017. Towards crafting text adversarial samples. *arXiv preprint arXiv:1707.02812*.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, 3104–3112.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I. J.; and Fergus, R. 2013. Intriguing properties of neural networks. *CoRR* abs/1312.6199.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Yang, P.; Chen, J.; Hsieh, C.-J.; Wang, J.-L.; and Jordan, M. I. 2018. Greedy attack and gumbel attack: Generating adversarial examples for discrete data. *arXiv preprint arXiv:1805.12316*.

Zhao, Z.; Dua, D.; and Singh, S. 2017. Generating natural adversarial examples. *arXiv preprint arXiv:1710.11342*.