Crafting Adversarial Examples for Neural Machine Translation

Xinze Zhang[1, 2, *] **Junzhe Zhang**[1] **Zhenhua Chen**[1] **Kun He**[1, †]
[1]School of Computer Science and Technology,
[2]School of Management,
Huazhong University of Science and Technology
{xinze,junzhezhang,zhenhuachen,brooklet60}@hust.edu.cn

Abstract

Effective adversary generation for neural machine translation (NMT) is a crucial prerequisite for building robust machine translation systems. In this work, we investigate veritable evaluations of NMT adversarial attacks, and propose a novel method to craft NMT adversarial examples. We first show the current NMT adversarial attacks may be improperly estimated by the commonly used monodirectional translation, and we propose to leverage the round-trip translation technique to build valid metrics for evaluating NMT adversarial attacks. Our intuition is that an effective NMT adversarial example, which imposes minor shifting on the source and degrades the translation dramatically, would naturally lead to a semantic-destroyed round-trip translation result. We then propose a promising black-box attack method called Word Saliency speedup Local Search (WSLS) that could effectively attack the mainstream NMT architectures. Comprehensive experiments demonstrate that the proposed metrics could accurately evaluate the attack effectiveness, and the proposed WSLS could significantly break the state-of-art NMT models with small perturbation. Besides, WSLS exhibits strong transferability on attacking Baidu and Bing online translators.

1 Introduction

Recent studies have revealed that neural machine translation (NMT), which has achieved remarkable progress in advancing the quality of machine translation, is fragile when attacked by some crafted perturbations (Belinkov and Bisk, 2018; Cheng et al., 2019, 2020; Wallace et al., 2020). Even if the perturbations on inputs are small and imperceptible to humans, the translation quality could be degraded dramatically, raising increasing attention to adversarial defenses for building robust machine translation systems as well as its prerequisite researches on building effective NMT adversarial attacks. As character level perturbations usually lead to lexical errors and are easily corrected by spell checking tools (Ren et al., 2019; Zou et al., 2020), in this work, we focus on crafting word level adversarial examples that could maintain lexical and grammatical correctness and hence are more realistic.

Input $\mathbf{x}$	John Biden just win the election
Trans. $\mathbf{y}$	约翰·拜登刚刚赢得了大选
Ref.	约翰·拜登刚刚赢得了选举
Input $\mathbf{x}'$	John Biden just lost the election
Trans. $\mathbf{y}'$	约翰·拜登刚刚赢得了大选

Table 1: A real example of adversarial generation for Google translation with antonym substitution (*i.e.*, *win* to *lost*) which reverses the semantics on the source but preserves the same translation exactly (reported in October, 2020).

An essential issue of crafting NMT adversarial examples is how to define "what is an effective NMT adversarial attack". Researchers have provided an intuitive definition that an NMT adversarial example should preserve the semantic meaning on the source but destroy the translation performance with respect to the reference translation (Michel et al., 2019; Niu et al., 2020). Correspondingly, the attack criteria are proposed as the absolute degradation or relative degradation against the reference translation (Ebrahimi et al., 2018; Michel et al., 2019; Niu et al., 2020; Zou et al., 2020). To craft a perturbation that maintains the semantics as well as grammatical correctness following the above definition and evaluation, a variety of methods to impose word replacements have been proposed in recent studies (Michel et al., 2019; Cheng et al., 2019, 2020; Zou et al., 2020), making it a commonly used paradigm for NMT attacks.

[*]The four authors contributed equally.
[†]Corresponding author: Kun He.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 1967–1977
August 1–6, 2021.

Reference Sentence	Chinese→English Translation
Ref.: *The chairperson of the conference expressed in a speech that high and new technologies have promoted the development of the nations in asia, europe, and america.*	$\mathbf{x}$: 会议主席在发言中认为, 高新技术促进了亚洲和欧美国家的发展。 $\mathbf{y}$: In his speech, the chairman of the meeting held that high and new technologies have promoted the development of asian and european countries.
Ref.$'_{\times}$: *The chairperson of the conference expressed in a speech that the high-level leadership has promoted the growth of the nations in asia, europe, and america.*	$\mathbf{x}'_{\times}$: 会议主席在发言中称, 高层促进了亚洲和欧美国家的成长。 $\mathbf{y}'_{\times}$: In his speech, the chairman of the meeting said that the high-level leadership has promoted the growth of asian and european countries.
Ref.$'_{\surd}$: *The chairperson of the convention expressed in a speech that the high-level leadership has promoted the development of the nations in asia, europe, and america.*	$\mathbf{x}'_{\surd}$: 代表大会主席在发言中称, 高层促进了亚洲和欧美国家的发展。 $\mathbf{y}'_{\surd}$: In his speech, the chairman of the npc standing committee said that the high-level leadership has promoted the development of asian and european countries.

Table 2: Two examples of adversarial generation for RNNsearch based NMT model with synonym substitution. The left column contains the ground-truth references. The right column contains the corresponding original input $\mathbf{x}$, noneffective adversarial example $\mathbf{x}'_{\times}$, effective adversarial example $\mathbf{x}'_{\surd}$, and their neural translations. The effective and noneffective attack locations are marked in orange and blue, respectively.

However, there exist potential pitfalls overlooked in existing researches. First, it is possible to craft an effective attack on the NMT models by reversing the semantics on the source, as illustrated in Table 1[1]. Meanwhile, since the antonyms are potentially in the neighborhood of the victim word in the embedding space, just as the same as the synonyms, it is entirely possible to produce opposing semantics when replacing a word with its neighbors, making the proposed attack method break the definition.

Furthermore, there is a risk of evaluating the attacks directly using the reference translation. Differs to the classification tasks, even if the perturbation is small to be synonymous with the original word in the source, the actual ground-truth reference may be changed due to the substitution. Table 2 illustrates a typical failing adversarial example $\mathbf{x}'_{\times}$ and a successful example $\mathbf{x}'_{\surd}$, where $\mathbf{x}'_{\times}$ could be falsely distinguished as effective due to the missing of ground-truth reference Ref.$'_{\times}$[2]. Obviously, $\mathbf{x}'$ would be correctly distinguished if we have the actual ground-truth reference of $\mathbf{x}'$. However, the actual ground-truth reference of the perturbed input is notoriously difficult to be built beforehand, making the NMT attack hardly to be evaluated veritably.

In this work, in order to craft appropriate NMT adversarial examples, we introduce new definition and metrics for the machine translation adversaries by leveraging the round-trip translation, the process of translating text from the source to target language and translating the result back into the source language. Our intuition is that an effective NMT adversarial example, which imposes minor shifting on the input and degrades the translation dramatically, would naturally lead to a semantic destroying round-trip translation result. Based on our new definition and metrics, we propose a promising black-box attack method called Word Saliency speedup Local Search (WSLS) that could effectively attack the mainstream NMT architectures, *e.g.* RNN and Transformer.

Our main contributions are as follows:

- We introduce an appropriate definition of NMT adversary and the deriving evaluation metrics, which are capable of estimating the adversaries only using source information, and tackle well the challenge of missing ground-truth reference after the perturbation.
- We propose a novel black-box word level NMT attack method that could effectively attack the mainstream NMT models, and exhibit high transferability when attacking popular online translators.

2 NMT Adversary Generation

Let $\mathcal{X}$ denote the source language space consisting of all possible source sentences and $\mathcal{Y}$ denote the target language space. Given two NMT models, the primal source-to-target NMT model $M_{x\to y}$ aims to

[1]This is a real case reported on Google translation community in October, 2020. See details in: https://support.google.com/translate/thread/78771708?hl=en.

[2]BLEU$(\text{ref}, \mathbf{y}) = 39.20 \rightarrow$ BLEU$(\text{ref}, \mathbf{y}'_{\times}) = 2.86$, BLEU$(\mathbf{x}, \mathbf{x}'_{\times}) = 61.34 \rightarrow$ BLEU$(\mathbf{y}, \mathbf{y}'_{\times}) = 49.83$.

learn a forward mapping $f : \mathcal{X} \to \mathcal{Y}$ to maximize $P(y_{ref}|x)$ where $x \in \mathcal{X}$ and $y_{ref} \in \mathcal{Y}$, while the dual target-to-source NMT model $M_{y \to x}$ aims to learn the backward mapping $g : \mathcal{Y} \to \mathcal{X}$. After the training, NMT can correctly reconstruct the source sentence $\hat{x} = g(f(x))$. In the following, we first give the definition of NMT adversarial examples, then introduce our word substitution based black-box adversarial attack method.

2.1 Definition on NMT Adversarial Examples

Given a subset of (test) sentences $\mathcal{T} \in \mathcal{X}$ and a small constant ϵ, we summarize previous works (Belinkov and Bisk, 2018; Ebrahimi et al., 2018; Michel et al., 2019) and give their conception of NMT adversarial examples as follows.

Definition 1 *(NMT Adversarial Example). An NMT adversarial example is a sentence in* $\mathcal{A} = \{x' \in \mathcal{X} | \exists x \in \mathcal{T}, \|x' - x\| < \epsilon \wedge S_t(y, y_{ref}) \geq \gamma \wedge S_t(y', y_{ref}) < \gamma'\}$, *where* $y = f(x), y' = f(x')$, *and* $S_t(\cdot, \cdot)$ *is a metric for evaluating the similarity of two sentences, and* γ *(or* γ', $\gamma' < \gamma$*) is threshold we can accept (or refuse) for the translation quality .*

A smaller γ' indicates a more strict definition of the NMT adversarial example.

In contrast to the adversarial examples in image domain (Szegedy et al., 2014), we argue that taking y_{ref} as the reference sentence for x' is not appropriate because the perturbation might change the semantic of x to some extent, causing that Definition 1 is not appropriate. To address this problem, we propose to evaluate the similarity between the benign sentence x and the reconstructed sentence $\hat{x}$, as well as the similarity between the adversarial sentence x' and the reconstructed adversarial sentence $\hat{x}'$. We introduce a new definition of NMT adversarial example basing on the round-trip translation.

Definition 2 *(NMT adversarial example). An NMT adversarial example is a sentence in* $\mathcal{A} = \{x' \in \mathcal{X} | \exists x \in \mathcal{T}, \|x' - x\| < \epsilon \wedge S_t(y, y_{ref}) \geq \gamma \wedge S_t(x, \hat{x}) \geq \delta \wedge E(x, x') \geq \alpha\}$, *where* $E(x, x') = S_t(x, \hat{x}) - S_t(x', \hat{x}')$ *is defined as the adversarial effect for NMT. And, the reconstructed* $\hat{x}$ *and* $\hat{x}'$ *are generated with round-trip translation:* $\hat{x} = g(f(x)), \hat{x}' = g(f(x'))$.

A larger E indicates that the generated sentence x' can not be well reconstructed by round-trip translation when compared with the reconstruction quality of the source sentence x. Here α is a threshold ranging in $[0, 1]$ to determine whether x' is an NMT adversarial example. A larger α indicates a more strict definition of the NMT adversarial example. In this work, we use the BLEU score (Papineni et al., 2002) to evaluate the similarity between two sentences.

Based on Definition 2, we further provide two metrics, *i.e.*, *Mean Decrease* (MD) and *Mean Percentage Decrease* (MPD) to estimate the translation adversaries appropriately. MD directly presents the average degradation of the reconstruction quality, and MPD reduces the bias of the original quality in terms of the relative degradation. The proposed MD is defined as:

$$MD = \frac{1}{N} \sum_{i}^{N} D_i, \tag{1}$$

where N is the number of victim sentences, D_i is the decreasing reconstruction quality of the adversarial example x'_i, denoted as:

$$D_i = \begin{cases} 0 & \text{if } S_t(x_i, \hat{x}_i) = 0, \\ S_t(x_i, \hat{x}_i) - S_t(x'_i, \hat{x}'_i) & \text{otherwise.} \end{cases} \tag{2}$$

Similarly, MPD is defined as:

$$MPD = \frac{1}{N} \sum_{i}^{N} PD_i, \tag{3}$$

where PD_i is denoted as:

$$PD_i = \begin{cases} 0 & \text{if } S_t(x_i, \hat{x}_i) = 0, \\ \frac{S_t(x_i, \hat{x}_i) - S_t(x'_i, \hat{x}'_i)}{S_t(x_i, \hat{x}_i)} & \text{otherwise.} \end{cases} \tag{4}$$

In practice, except for the constraints in Definition 2, adversarial examples should also satisfy the lexical and syntactical constraints so that they are hard for human to perceive. Therefore, the correct word in the source sentence must be replaced with other correct words instead of misspelled word to meet the lexical constraint. Besides, to keep the grammatical correctness and syntax consistency, the modification should not change the syntactic relation of each word in the source sentence.

To meet all the above constraints, we propose a novel NMT adversarial attack method by substituting words with their neighbors selected from the parser filter to generate reasonable and effective adversarial examples.

2.2 WSLS Attack

There are two phases in the proposed Word Saliency speedup Local Search (WSLS) attack

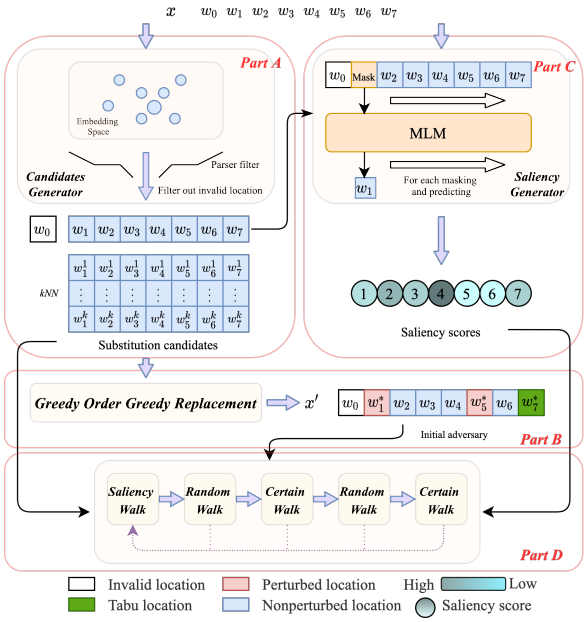


Figure 1: Illustration of the proposed WSLS attack method. For a source sentence x, we first generate the valid victim locations, substitution candidates, and saliency scores to prepare the attack, then craft an initial adversarial example x' by the Greedy Order Greedy Replacement (GOGR) followed by the Word Saliency speedup Local Search (WSLS) to promote the adversarial quality.

method. At the first phase, we design initial strategies to obtain an initial example x'. At the second phase, we present a local search algorithm accelerated by word saliency to optimize the perturbed example.

2.2.1 Initialization Strategy

Candidates. For a word w_i in the source sentence $x = \{w_1, \ldots, w_i, \ldots, w_n\}$, where i denotes the position of word w_i in the sentence, we first build a candidate set $\mathbb{W}_i \in \mathbb{D}$ where $\mathbb{D}$ is the dictionary consisting of all the legal words. In this work, we build the candidate set by finding the k closest neighbors in the word embedding space: $\mathbb{W}_i = \{w_i^1, \ldots, w_i^k\}$. Then we filter the candidates based on the parsing, as shown in *Part A* of Figure 1[3]. Note that the combination of them can impose minor shifting on the source so as to meet the lexical and semantic constraints, as discussed in Section 2.1. In our experiments, we use the pre-trained mask language model (MLM) to extract the embedding space to follow the black-box setting.

[3]This is important to rule out invalid victim locations wherein the token (*e.g.*, punctuation) is nonsense, and ensure the perturbations keep grammatical correctness.

Greedy Substitution. For each position i, we can substitute word w_i with $w_i^j \in \mathbb{W}_i$ to obtain an adversary $x' = \{w_1, \ldots, w_i^j, \ldots, w_n\}$, and evaluate the adversarial effect $E(x, x')$ by reconstruction. Then we select a word w_i^* that yields the most significant degradation:

$$w_i^* = \arg\max_{w_i^j \in \mathbb{W}_i} E(x, x'). \quad (5)$$

It is straightforward to generate an initial adversary through a *Random Order Greedy Replacement* (ROGR) method, which is to randomly select positions expected to make substitutions, then iteratively replace the word with its neighbors by Eq. 5 on the selected positions in a random order.

However, the initial result has a significant impact on the final result of the local search. If the local search phase starts with a near-optimal solution, it is likely to find a more powerful adversary after the local search process. Therefore, we design a greedy algorithm called *Greedy Order Greedy Replacement* (GOGR) for the initialization, which is depicted in *Part B* of Figure 1.

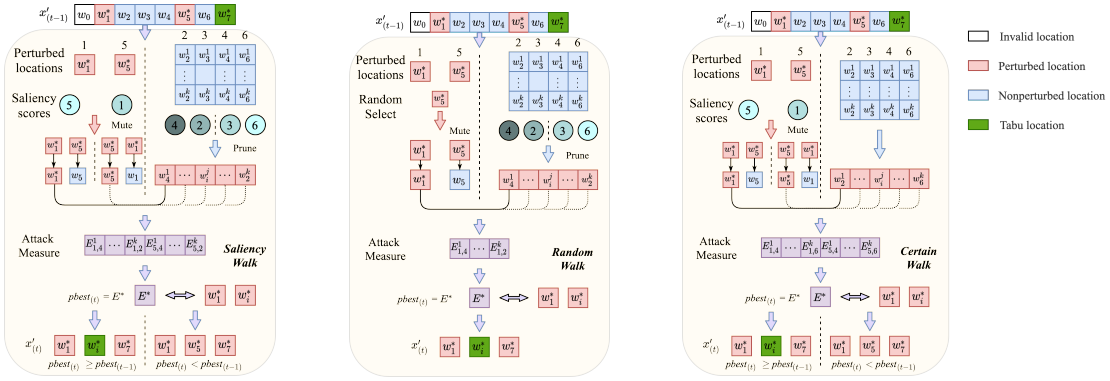
In the GOGR algorithm, at each step we enumerate all possible positions we haven't attacked yet, and for each position we try to substitute word $w_i \in x$ with word $w_i^* \in \mathbb{W}_i$ according to Eq. 5, then we choose the best w^* among the possible positions, and iteratively substitute words until we substitute enough words.

$$w^* = \arg\max_{i \in n} \max_{w_i^j \in \mathbb{W}_i} E(x, x') \quad (6)$$

2.2.2 Word Saliency

To speed up the local search process, we adopt the *word saliency*, used for text classification attack, to sort the word positions in which the word has not been replaced yet. In this way, we can skip the positions that may lead to low attack effect so as to speedup the search process.

For text classification task, Li et al. (2016) propose the concept of word saliency that refers to the degree of change in the output of text classification model when a word is set to the "unknown" token. Ren et al. (2019) incorporate the word saliency to generate adversarial examples for text classification. To adopt the concept of word saliency for NMT, we regard the output of a MLM for the word as a more general concept of word saliency, which is independent of the specific tasks.



(a) Saliency Walk (b) Random Walk (c) Certain Walk

Figure 2: Illustration of the walks used in the local search.

Definition 3 *(Word Saliency). For a sentence* $x = \{w_1, \ldots, w_i, \ldots, w_n\}$ *and a mask language model (MLM)* M*, the word saliency of* w_i *is defined as* $S(x, w_i) = 1 - P(w_i|\bar{x}_i, M)$ *where* $\bar{x}_i = \{w_1, \ldots, w_{i-1}, mask, w_{i+1} \ldots, w_n\}$ *and "mask" means the word is masked in the sentence.*

Through Definition 3, the higher word saliency represents the lower context-dependent probability, which can be caused by numerous reasonable substitutions or rare syntax structure, indicating weaker word positions that are easier to be attacked.

In this work, as shown in *Part C* of Figure 1, we calculate the word saliency $S(x, w_i)$ for all positions before the local search phase, making the local search efficiently inquire the word saliency.

2.2.3 Local Search Strategy

In the local search phase, as shown in *Part D* of Figure 1 and detailed in Figure 2, there are three types of walks, namely *saliency walk*, *random walk* and *certain walk*, used to update x' to promote the attack quality.

To explore and exploit the search space, we define some basic operations and walks to evolve the adversaries. A **mute** operator is to restore an executed perturbation w_i^* to its original word w_i to mutate the adversary. A **prune** operator is to exclude a portion of candidate locations where the perturbations will not be imposed to narrow down the search area. A **tabu** operator indicates that the last perturbed location is forbidden to be manipulated in the current iteration. As illustrated in Figure 2, the three operators are utilized in the local search walks (*Part D*). We interpret the three walks as follows.

Saliency Walk. We first design an efficient walk for the search, called the saliency walk (SW), to make a balanced exploration and exploitation in the neighbourhood of the well initialized solution generated by the aforementioned GOGR algorithm. During the saliency walk, as shown in Figure 2a, at the current iteration (t), we mute each perturbed word to generate a set of partial solutions, sorted in the ascending order of the saliency score, so as to give higher priority to the perturbations with higher word saliency on the locations. Then we prune other unperturbed words according to the descending order of the saliency score, and query candidate substitutions for each of the remaining words. Then candidate adversaries, consisting of the concatenation of each partial solution with each candidate substitution, are evaluated by Eq. 2 iteratively.

To accelerate the saliency walk, we have an early stop strategy: if the current best adversarial effect in the enumeration of the candidate adversaries at the present iteration (t), denoted as $pbest_{(t)} = E^*$, is better than $pbest_{(t-1)}$ (the best adversarial effect at the previous iteration $(t-1)$), *i.e.* $pbest_{(t)} \geq pbest_{(t-1)}$, then we terminate the enumeration of the candidates and pass the state of $pbest_{(t)}$ as well as the tabu operator to the next walk, otherwise the state of $pbest_{(t-1)}$ will be passed to the next walk and the tabu location is expired.

Random Walk. To avoid the current adversarial example get trapped in a local optimum, we design an effective mutation walk, called the random walk (RW), to mutate the current solution. During the random walk, as shown in Figure 2b, we randomly mute a perturbed word to generate a partial solution, and query the candidate substitutions for each of the unperturbed words as in saliency walk. Then we concatenate the partial solution with each candidate substitution to build the candidate adver-

saries, among which the best solution is used to update $pbest_{(t)}$. After that, the tabu operator will be forcibly passed to the next walk, reinforcing the exploration ability of the WSLS algorithm.

Certain Walk. To do a sufficient exploitation after the random walk as a mutation, we design the certain walk (CW). As shown in Figure 2c, certain walk is similar to saliency walk but it removes the prune operation to enlarge the neighborhood space.

To trade off the efficiency and search time, we adopt one saliency walk followed by random walk, certain walk, random walk and certain walk, to construct one round of local search, denoted as {SW, RW, CW, RW, CW}, as shown in *Part D* of Figure 1. Besides, we bring an early-stop-finetune mechanism to the WSLS method. For any walk in WSLS, if there exists an adversarial candidate that updates the historically best adversarial effect, this adversarial candidate will be immediately set as the initial solution to start a new local search. Otherwise, the WSLS will stop after the ending of the current round [4].

3 Experiments

3.1 Experimental Setup

We conduct experiments on the Chinese-English (Zh-En), English-German (En-De), and English-Russian (En-Ru) translation tasks.

For the Zh→En translation task, we use LDC corpus[5] consisting of 1.25M sentence pairs, and use NIST (MT) datasets[6] to craft the attacks. Following the preprocessing in Zhang et al. (2019), we limit the source and target vocabulary to the most frequent 30K words, remove sentences longer than 50 words from the training data, and use NIST 2002 as the validation set for the model selection. For this translation task, we implement our attacks on two state-of-art word-level NMT models. 1) **RNNsearch** (Bahdanau et al., 2015) has an encoder consists of forward and backward RNNs each having 1000 hidden units and a decoder with 1000 hidden units. Denote this model as "*Rnns.*" for abbreviation. 2) **Transformer** comprises six layers of transformer with 512 hidden units and 8 heads in both encoder and decoder, which mimics the hyperparameters in (Vaswani et al., 2017). Denote this model as "*Transf.*" for abbreviation. For the oracle back-translation (En→Zh), we use a sub-word level transformer as our oracle model which was trained with LDC datasets and then finetuned with the NIST datasets.

For the En→De and En→Ru translation tasks, We use WMT19 test sets to craft the adversaries, and implement our attacks on the winner models of the WMT19 En→De and En→Ru sub-tracks[7]. Specifically, the En→De model and En→Ru model are both subword-level transformer, where a joint byte pair encodings (BPE) with 32K split operations is applied for En→De, and separate BPE encodings with 24K split operations is applied for each language in En→Ru (Ng et al., 2019). We denote these two models as "*BPE-Transf.*" for abbreviation. For the oracle back-translation (De→En, Ru→En), the best submitted NMT models in WMT19 are used as our oracle models which are further finetuned with 90% of the previous WMT test sets and validated with the remaining sets.

As for the reference result, Table 3 and Table 4 show the case-insensitive BLEU scores for forward-translation, back-translation, and round-trip translation on the selected language pairs. We observe that the word-level victim models (*Rnns.* and *Transf.*) achieve an average BLEU score of 36.71 and 41.55 for Zh→En translation respectively, demonstrating the accuracy of these two models on translating the original Chinese sentences. For the back-translation, the oracle models achieve an average BLEU score of 82.9 for En→Zh translation, as well as a BLEU score of 54.83 and 57.24 for De→En and Ru→En translations respectively, indicating that the oracle models are reliable enough in the back-translation stage for the source reconstruction. Besides, the reconstruction quality of the victim models are reported in Table 3 and Table 4, where the source sentences are back-translated by the oracle models in the round-trip translation, showing that the source language is reconstructed well enough by the cooperation of forward-translation and oracle back-translation.

Furthermore, to enhance the authenticity of the attack performance, we removed the noisy data, which could not be correctly identified as the corresponding language sentences by online translators, and we also excluded sentences longer than 50 words in the NIST datasets, ensuring that the attack

[4]Code is available at https://github.com/JHL-HUST/AdvNMT-WSLS/.

[5]LDC 2002E18, 2003E14, 2004T08, 2005T06.

[6]NIST 2002, 2003, 2004, 2005, 2006, 2008.

[7]https://github.com/pytorch/fairseq/tree/master/examples/translation.

Translation	Model	MT02	MT03	MT04	MT05	MT06	MT08	AVG
Forward	*Rnns.*	40.07	37.42	40.30	37.48	36.52	28.48	36.71
	Transf.	43.70	42.31	44.25	42.73	42.22	34.06	41.55
Back	*Oracle*	88.63	84.55	79.14	80.69	85.26	79.34	82.94
Round-trip	*Rnns.*	55.46	44.43	55.27	44.97	46.99	36.91	47.34
	Transf.	70.90	59.62	68.44	60.92	61.78	51.06	62.12

Table 3: Case-insensitive BLEU scores (%) for forward-translation (Zh→En), back-translation (En→Zh), and round-trip translation (Zh→En→Zh) on Zh-En language pair. "AVG" represents the average score of all datasets.

Language pair	Translation		
	Forward	Back	Round-trip
En-De	46.29	56.19	61.87
En-Ru	47.23	58.16	57.60

Table 4: Case-insensitive BLEU scores (%) of *BPE-Transf.* for forward-translation, back-translation, and round-trip translation on En-De and En-Ru language pairs.

results are credible[8].

As for the parameter settings of the attack methods, we use pyltp[9] as the parser checking tool and generate the top 10 nearest parser-filtered words to construct the candidate sets for each word. To generate the word saliency, two state-of-art whole word masking BERT are utilized as the MLM for the Chinese[10] and English[11] languages respectively. And the prune operators implemented in SW and RW will reserve the highest five word saliency locations and their word candidates. Finally, the adversaries are crafted by substituting 20% words.

3.2 Attack Results

To demonstrate our proposed WSLS method, we implement AST-lexcial (Cheng et al., 2018) as a black-box baseline, wherein AST-lexcial shares the same idea of random order random replacement. Besides, the naive ROGR method can be considered as another black-box counterpart of the white-box kNN method in Michel et al. (2019) that randomly selects the word positions and greedily selects the neighbor words based on the gradient loss.

As shown in Table 5 and Table 6, both GOGR and WSLS have the MD scores close to the original reconstruction scores for *Rnns.*, *Transf.*, and *BPE-Transf.*, and their attack results are much better than that of AST-lexical as well as ROGR. It shows that both WSLS and GOGR can effectively attack various NMT models under the standard of Definition 2. WSLS is superior to GOGR, indicating that the local search phase can further promote the attack quality. Specifically, the MPD score of WSLS is almost 1.5 higher than that of GOGR, which is more obvious as compared to the MD metric, revealing the rationality of MPD also.

3.3 Ablation Study

We do ablation study on the WSLS algorithm in Table 7. Here "Init" is for the method used for initialization, WS indicates whether we use word saliency to speedup the local search, LS indicates whether we use local search or other variants of walk sequence for the local search.

From Table 7 we observe that: 1) The initialization of GOGR exhibits significantly better results than ROGR, and also converges faster than ROGR; 2) WSLS without word saliency speedup, denoted as WSLS_1, exhibits slightly higher attack results but the running times are much longer than WSLS. Thus, we choose WSLS to have a good tradeoff on attack quality and time.

3.4 Transferability

To test the transferability of our method, we transfer our crafted adversarial examples on NIST 2002 dataset to attack the online Baidu and Bing translators. As shown in Table 8, the attack effectiveness is significant. It degrades the reconstruction quality of Baidu and Bing with more than 20 BLEU points, demonstrating the high transferability.

In addition, we provide two adversarial examples in Table 9, generated by WSLS on the *Rnns.* model, that can effectively attack the online Bing

[8] After the preprocessing, the size of the original NIST datasets are reduced from 878 to 617 (MT02), 919 to 793 (MT03), 1788 to 1495 (MT04), 1082 to 907 (MT05), 1664 to 988 (MT06), and 1357 to 789 (MT08).

[9] https://github.com/HIT-SCIR/pyltp.

[10] https://huggingface.co/hfl/chinese-bert-wwm-ext.

[11] https://huggingface.co/bert-large-uncased-whole-word-masking.

Metrics	Model	Method	MT02	MT03	MT04	MT05	MT06	MT08	AVG
MD	*Rnns.*	AST-lexical	29.06	23.61	29.16	23.83	26.42	20.43	25.78
		ROGR	38.54	30.64	38.04	32.00	33.34	26.13	33.12
		GOGR	51.09	39.78	50.72	41.61	42.82	32.74	43.13
		WSLS	**51.51**	**40.15**	**51.19**	**41.96**	**42.84**	**33.03**	**43.45**
	Transf.	AST-lexical	38.65	32.78	36.72	32.92	35.72	28.88	34.28
		ROGR	48.09	41.88	45.75	42.27	43.64	36.14	42.96
		GOGR	65.34	54.96	62.85	56.91	56.52	45.25	56.97
		WSLS	**66.03**	**55.47**	**63.51**	**57.39**	**57.02**	**45.69**	**57.51**
MPD	*Rnns.*	AST-lexical	51.24	44.41	42.21	39.66	42.59	42.2	43.17
		ROGR	70.42	70.71	69.50	67.27	69.00	69.26	69.36
		GOGR	93.96	93.08	94.08	92.62	91.92	90.12	92.63
		WSLS	**95.18**	**94.23**	**95.17**	**93.68**	**93.11**	**91.80**	**93.86**
	Transf.	AST-lexical	52.25	45.03	50.88	48.91	51.40	43.32	48.63
		ROGR	70.36	70.42	69.01	69.50	72.17	73.01	70.75
		GOGR	95.25	94.45	94.81	95.09	93.94	92.72	94.38
		WSLS	**96.24**	**95.69**	**95.97**	**96.27**	**95.08**	**94.32**	**95.60**

Table 5: MD and MPD results (%) on *Rnns.* and *Transf.* attacked by various methods on the preprocessed NIST datasets. A higher result indicates a better attack method.

Metrics	Task	Method			
		AST-lexical	ROGR	GOGR	WSLS
MD	En-De	26.47	38.78	52.85	**54.57**
	En-Ru	28.02	36.59	49.21	**49.92**
MPD	En-De	42.77	64.19	90.74	**92.22**
	En-Ru	42.96	66.56	91.15	**92.48**

Table 6: MD and MPD results (%) on *BPE-Transf.* attacked by various methods on WMT19 test sets.

Method	Init	WS	LS	Time	MD	MPD
ROGR	rogr	✗	✗	0.34	48.09	70.36
GOGR	gogr	✗	✗	2.87	65.34	95.25
$WSLS_1$	gogr	✗	R+C	25.47	**67.10**	**96.60**
$WSLS_2$	rogr	✗	R+C	33.27	65.62	94.81
WSLS	gogr	✓	Std.	8.23	66.03	96.24

Table 7: The ablation study on *Transf.* with ablative algorithms (R, C, Std. indicate random walk, certain walk and standard WSLS algorithm) on MT02 dataset. The running time is in minutes per sentence.

and Baidu translators, respectively. It demonstrates that WSLS could craft adversarial examples with strong readability and high transferability.

4 Related Work

In recent years, adversarial examples have attracted increasing attention in the area of natural language processing (NLP), mainly on text classification (Jia and Liang, 2017; Ren et al., 2019; Wang et al., 2021). For neural machine translation (NMT), there are also some adversary works emerging quickly (Belinkov and Bisk, 2018; Ebrahimi et al., 2018; Michel et al., 2019; Cheng et al., 2019; Niu et al., 2020; Wallace et al., 2020).

Transfer	Victim	Methods		
		ROGR	GOGR	WSLS
Baidu	*Rnns.*	19.96	18.30	18.61
(41.81)	*Transf.*	19.64	16.89	17.30
Bing	*Rnns.*	17.59	15.29	15.51
(38.15)	*Transf.*	17.13	14.82	14.68

Table 8: Reconstruction quality on Baidu and Bing online translators for the adversaries generated on the Zh→En task using MT02 dataset, wherein the adversaries are reconstructed by the online translators (Zh→En) and oracle (En→Zh). By contrast, the benign reconstruction quality is in the bracket.

On the character level, a few adversarial attacks by manipulating character perturbations have been proposed since 2018. Belinkov and Bisk (2018) confront NMT models with synthetic and natural misspelling noises, and show that character-based NMT models are easy to be attacked by character level perturbation. Ebrahimi et al. (2018) propose to attack the character level NMT models by manipulating the character-level insertion, swap and deletion. Similarly, Michel et al. (2019) perform a gradient-based attack that processes words in source sentences to maximize the translation loss. To attack against production MT systems, Wallace et al. (2020) imitate the popular online translators and manipulate the perturbations based on the gradient of the adversarial loss with the imitation models. The above four works also incorporate adversarial training to improve the robustness of NMT.

However, the character level perturbations are hard to be applied into confronting practical NMT models, as these perturbations significantly reduce

$\mathbf{x}'$: 代表大会主席在发言中称, 高层促进了亚洲和欧美国家的发展。 Ref.′: *The chairperson of the convention expressed in a speech that the high-level leadership has promoted the development of the nations in asia, europe, and america.* **Baidu**: In his speech, the president of the National People's Congress said that high-level leaders have promoted the growth of asian and european countries.
$\mathbf{x}'$: 彼得森重申, 世卫组织主要关切的难题是防止诸如疾病、痢疾等疫情暴发, 这些患者可能造成成千上万的人罹难 。 Ref.′: *Peterson reiterated that the WHO's main concern is the challenge of preventing outbreaks such as disease and dysentery, these patients may cause thousands of deaths.* **Bing**: Peterson reiterated that the WHO's main concern is to prevent outbreaks such as disease and dysentery , which can cause thousands of deaths.

Table 9: Two examples of attacking online translators, in which the adversaries are generated on the *Rnns.* model using WSLS.

the readability and also could be easily corrected by spell checkers (Ren et al., 2019; Zou et al., 2020). On the other hand, word level adversaries could maintain lexical and grammatical correctness, which are more realistic but more challenging to generate. Cheng et al. (2018) craft the adversaries with randomly sampled perturbed positions, and then replace the words according to the cosine similarity of the embedding vectors between the original word and the neighbors. Cheng et al. (2019) propose a gradient-based attack method that replaces the original word with the candidates generated by integrated language model. Michel et al. (2019) generate adversaries by substituting the word with its nearest neighbors, which are informed by the gradient of the victim models. (Zou et al., 2020) introduce a reinforced learning based method to craft the attacks following Michel et al. (2019) to define the reward and substitution candidate set.

Existing word level translation attacks are mainly white-box, wherein the attacker can access all the information of the victim model. Besides, there is a risk of guiding the attacks to directly use the degradation of reference translation, since the actual references may be changed by word substitution. Thus, there exists few study on the effective word level attack for NMT, especially in the black box setting. This study fills this gap and sheds light on black-box word level NMT attacks.

5 Conclusion

We introduce an appropriate definition of adversarial examples as well as the deriving evaluation measures for the adversarial attacks on neural machine translation (NMT) models. Following our definition and metrics, we propose a promising black-box NMT attack method called the Word Saliency speedup Local Search (WSLS), in which a general definition of word saliency by leveraging the strong representation capability of pre-trained language models is also introduced. Experiments demonstrate that the proposed method could achieve powerful attack performance, that effectively breaks the mainstream RNN and Transformer based NMT models. Further, our method could craft adversaries with strong readability as well as high transferability to the popular online translators.

Acknowledgements

This work is supported by National Natural Science Foundation (62076105) and Microsft Research Asia Collaborative Research Fund (99245180). We thank Xiaosen Wang for helpful suggestions on our work.

References

Dzmitry Bahdanau, KyungHyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations*.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine Translation. In *Proceedings of the International Conference on Learning Representations*.

Yong Cheng, Lu Jiang, and Wolfgang Macherey. 2019. Robust neural machine translation with doubly adversarial inputs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Yong Cheng, Lu Jiang, Wolfgang Macherey, and Jacob Eisenstein. 2020. AdvAug: robust data augmentation for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Yong Cheng, Zhaopeng Tu, Fandong Meng, Junjie Zhai, and Yang Liu. 2018. Towards robust neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

Marta R Costa-jussà and José AR Fonollosa. 2016. Character-based neural machine translation. In *Proceedings of the 54th Annual Meeting of the 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 North American Chapter of the Association for Computational Linguistics*.

Javid Ebrahimi, Daniel Lowd, and Dejing Dou. 2018. On adversarial examples for character-level neural machine translation. In *Proceedings of the International Conference on Computational Linguistics*.

Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tie-Yan Liu, and Wei-Ying Ma. 2016. Dual learning for machine translation. In *Proceedings of the Advances in Neural Information Processing Systems*.

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

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. In *Proceedings of the Advances in Neural Information Processing Systems*.

Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2016. Visualizing and understanding neural models in nlp. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*.

Paul Michel, Xian Li, Graham Neubig, and Juan Pino. 2019. On evaluation of adversarial perturbations for sequence-to-sequence models. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook fair's wmt19 news translation task submission. In *Proceedings of the 4th Conference on Machine Translation*.

Xing Niu, Prashant Mathur, Georgiana Dinu, and Yaser Al-Onaizan. 2020. Evaluating robustness to input perturbations for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the 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*.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the Advances in Neural Information Processing Systems*.

Eric Wallace, Mitchell Stern, and Dawn Song. 2020. Imitation attacks and defenses for black-box machine translation systems. In *Proceedings of Empirical Methods in Natural Language Processing*.

Xiaosen Wang, Yichen Yang, Yihe Deng, and Kun He. 2021. Adversarial training with fast gradient projection method against synonym substitution based text attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Wen Zhang, Yang Feng, Fandong Meng, Di You, and Qun Liu. 2019. Bridging the gap between training and inference for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Wei Zou, Shujian Huang, Jun Xie, Xinyu Dai, and Jiajun Chen. 2020. A reinforced generation of adversarial samples for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Appendix

In the appendix, we provide necessary background of Neural Machine Translation (NMT), pre-trained language models, and the back-translation technique used in related works. Besides, screenshots of Table 8 are also provided.

Neural Machine Translation. Typical NMT models follow an encoder-decoder architecture with attention mechanisms (Zhang et al., 2019). The encoder encodes the source language to a latent representation space, and the decoder is a neural language model that decodes representations in the latent space to another language domain. Either the encoder or the decoder can be built on recurrent neural networks (Bahdanau et al., 2015), convolutional neural networks (Costa-jussà and Fonollosa, 2016), or Transformer networks (Vaswani et al., 2017). In this work, we applied two versions of neural network architecture for the encoder/decoder models: RNN and Transformer.

Pre-trained Language Model. Recently, pre-trained language models, such as mask language

Figure 3: An example of attacking the baidu translator, in which the adversarial example is generated on the *Rnns.* model using WSLS.

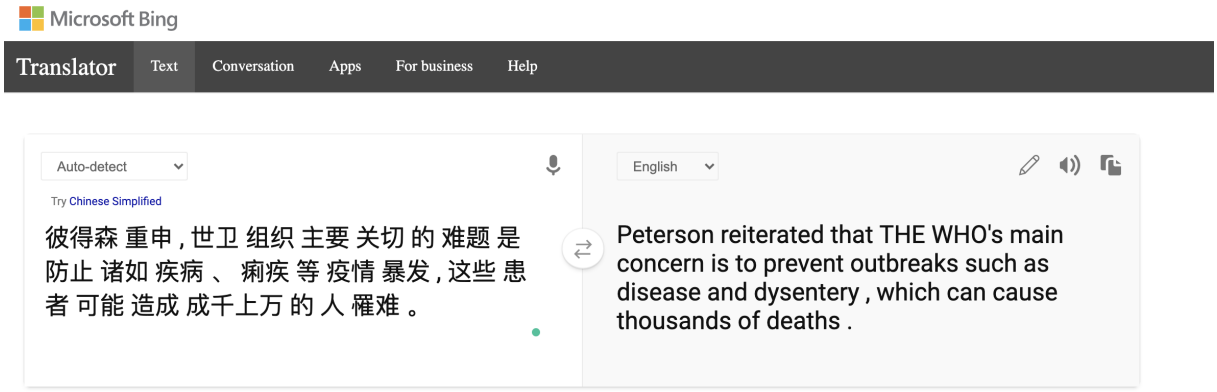


Figure 4: An example of attacking bing translator, in which the adversarial example is generated on the *Rnns.* model using WSLS.

models (MLM) (Devlin et al., 2019), have achieved a powerful initialization for the NMT encoder models. MLM pre-trains the encoder for a better language understanding on the encoded language by randomly masking some tokens in continuous monolingual text streams and predicting these tokens. To predict the masked tokens, the language model pays attention to the relative language parts, which encourages the model to have a better understanding on the language. Inspired by the powerful language understanding ability of the pre-trained language models, and following the black-box setting, we use the pre-trained MLM to estimate the word saliency and build the word embedding space for adversarial attacks.

Back-Translation. There are a lot of works for improving the NMT performance by leveraging the back translation, which uses not only parallel corpus but also monolingual corpus for training the NMT models (He et al., 2016; Lample and Conneau, 2019). Previous works on back-translation demonstrate the ability of the dual NMT models to reconstruct the language. In this work, we observe that the back-translation technique makes it possible to evaluate NMT adversarial attacks without ground-truth references for the perturbed sentences, and we propose to evaluate the proposed NMT attack method basing on the reconstruction results of the original inputs and the perturbed examples.