Machine Learning (2022) 111:3977–4002
https://doi.org/10.1007/s10994-022-06249-x

Attacking neural machine translations via hybrid attention learning

Mingze Ni[1] · Ce Wang[2] · Tianqing Zhu[1] · Shui Yu[1] · Wei Liu[1]

Received: 6 April 2022 / Revised: 28 July 2022 / Accepted: 13 September 2022 /
Published online: 20 October 2022

Abstract
Deep-learning based natural language processing (NLP) models are proven vulnerable to adversarial attacks. However, there is currently insufficient research that studies attacks to neural machine translations (NMTs) and examines the robustness of deep-learning based NMTs. In this paper, we aim to fill this critical research gap. When generating word-level adversarial examples in NLP attacks, there is a conventional trade-off in existing methods between the attacking performance and the amount of perturbations. Although some literature has studied such a trade-off and successfully generated adversarial examples with a reasonable amount of perturbations, it is still challenging to generate highly successful translation attacks while concealing the changes to the texts. To this end, we propose a novel Hybrid Attentive Attack method to locate language-specific and sequence-focused words, and make semantic-aware substitutions to attack NMTs. We evaluate the effectiveness of our attack strategy by attacking three high-performing translation models. The experimental results show that our method achieves the highest attacking performance compared with other existing attacking strategies.

Keywords Adversarial learning · Neural machine translation · Attention models

Editors: João Gama, Alípio Jorge and Salvador García.

✉ Wei Liu
wei.liu@uts.edu.au

Mingze Ni
mingze.ni@student.uts.edu.au

Ce Wang
wce@pku.edu.cn

Tianqing Zhu
tianqing.zhu@uts.edu.au

Shui Yu
shui.yu@uts.edu.au

[1] School of Computer Science, University of Technology Sydney, Sydney, Australia

[2] Peking University, Beijing, China

1 Introduction

Natural language processing (NLP) models are crucial for numerous AI-related applications, including sentiment analysis (Li et al., 2021; Xue et al., 2022), knowledge tracing (Song et al., 2021, 2022), question answering (Berant et al., 2013), and machine translation (Luong et al., 2017; Dzmitry Bahdanau & Bengio, 2015). These models exploit contextual information in textual sequences which make them vulnerable to text perturbation. Among the NLP tasks, NMTs can also be sensitive to adversarial examples, such as malicious tampering and input typos, as the sequence-to-sequence mapping relies on both the accuracy of individual word translation and contextual correlation within a sentence. Therefore, as a practical application that can be broadly applied for commercial purposes, the robustness of NMTs against adversarial attacks is highly desired, posing the necessity of studying NMT-targeted attacks.

Existing attack methods to NLP models can be generally divided into character-level and word-level attacks. Character-level attacks, which manipulate informational letters within a word to attack the victim NLP model with incorrectly spelled examples, have been explored and proven effective in both white-box and black-box settings (Belinkov & Bisk, 2017; Ebrahimi et al., 2018). However, character-level attacks can be easily defended by spelling auto-correction methods. In contrast, word-level attack methods hold that an adversary should locate the vulnerable words and manipulate them, such as swapping, inserting, deleting, and substituting, to deceive the NLP models (Cheng et al., 2019; Alzantot et al., 2018). However, word-level attacks to NLP models usually have a trade-off where the attacking performance depends on the number of perturbed words (Michel et al., 2019). Despite of the constant efforts on improving NLP attack methods, it is still challenging to strike such a balance between the number of perturbed words and its effectiveness in existing works, which is particularly true for attacks to NMTs which have not been well studied in the literature.

To this end, we argue that it is necessary to have an attack method that maximizes the attacking performance without having to increase the number of word perturbations. Therefore, we propose an attack strategy with a two-step approach: (1) a hybrid attention attack strategy to locate the top vulnerable words (i.e., victim words). This strategy consists of two types of attention weights: a language-specific attention that examines the correlation of words between source and target languages, and a sequence-centered self-attention that focuses on the language understanding of the source sentence itself. (2) a pre-trained Mask Language Model (MLM) to make semantic-aware substitutions to the victim words discovered in (1), to ensure that the generated adversarial examples are semantically correct. With the proposed strategy, we can make high-quality word-level attacks to NMTs with only a small amount of perturbations.

Specifically, the main contributions of this paper are as follows:

- We propose a novel Hybrid Attentive Attack (HAA) method which identifies the most influential words in an input sequence based on language-specific and sequence-centered attentions.
- We introduce a semantic-aware word substitution strategy for the proposed HAA method to strike a balance between attack effectiveness and imperceptibility.
- We conduct extensive experiments on real-world datasets with three state-of-the-art victim NMTs. Experimental results demonstrate that our proposed method achieves the best performance with a small number of perturbed words.

2 Related Work

In this section, we will introduce some previous about the textual attacks to NLP model, attention mechanisms and BERT variants.

2.1 Word-level attacks to NLP models

Word-level attacks pose non-trivial threats to NLP models by locating the victim words and manipulating them for targeted or untargeted purposes. With the help of an adopted FGSM (Goodfellow et al., 2015), Papernot (2016) was the first one to generate word-level adversarial examples to classifiers. They replaced the randomly chosen words and find the substitution with the help of the gradient to pose adversarial threat. Notably, while the textual data is naturally discrete, many gradient-based victim words selection methods are inherited from computer vision (Chivukula & Liu, 2018; Yin et al., 2018; Chivukula & Liu, 2017), which leaves locating victim words a challenging problem (Yang et al., 2020, 2021). Many existing methods randomly select the victim words, without considering the gradient and contextual information, and focus on words manipulations (Zang et al., 2020; Cheng et al., 2020; Wang et al., 2021) while Liang argues the selection of victim words is also important. To concrete this, he performed a white-box attack they provided a concept of Hot Training Phrase (HTP) and Hot Sample Phrase (HSP) to select the victim words with the help of backpropagation to get all the cost gradients (Liang et al., 2017). To make a more practical black-box setting, Gao (2018) proposed a new criterion without gradient information for locating the victim words to attack classifiers, by greedily searching the word with the highest score on the criterion. Furthermore, Li (2020) defined a score function by applying the logits from BERT (Devlin et al., 2019) for selecting the victim words, and then substitute them with BERT to attack downstream jobs based on BERT.

NMT, a type of NLP models, is an approach to machine translation that uses an deep learning techniques to predict the likelihood of a sequence of words, typically modeling entire sentences in a single integrated model (Kalchbrenner & Blunsom, 2013). Since the NMT is based on deep learning techniques and can be used for commercial purposes, there are raising number of researchers concern that the security and fairness of NMT can be abused. The attack for NMT is firstly introduce by Belinkov (2017), who worked with character-based neural machine translation and tent to attack NMT with natural typos without assuming any gradients. In addition to the attacks, they have explored two approaches to increase model robustness: structure-invariant word representations and robust training on noisy texts. Ebrahimi (2018) provided white and black box attack techniques and showed that white-box attacks were more damaging than black-box attacks, while black-box setting is more practical. For white-box attack, they tried to mute or push a particular word in a translation task by using gradient-based optimization. As for black-box attack, they just randomly picked a character and made necessary changes. Different from the two previous pioneers, Cheng (2019) proposed a gradient-based white-box attack technique called AdvGen to attack NMT in sentence-level. Guided by the training loss they used a greedy choice based approach to find the best solution. Their research paper is based on using adversarial examples for both attack generation and using these adversarial examples to improve the robustness and security of the model. While Michael (2019) also worked on textual white-box attacks to NMTs from a sentence-level and proposed a natural criterion for untargeted attacks. They argued that adversarial examples should be meaning preserving on the source

side but meaning destroying on the target side. They used the gradients of the model which replaces one word from the sentences to maximize the loss while they used KNN to determine the top 10 words which are similar to the victim word for purpose of preserving the semantic means. Besides, it was also proposed to attack NMTs via data poisoning (i.e., changing the training data) (Xu et al., 2021).

The pre-mentioned attacking strategies are all from character and word levels, while they all have some drawbacks such as being detected word correction system, too perceptible for human eyes. Different from these pioneers who attacked the NMT from character-level and sentence-level, Tan (2020) proposed to attack NMT in a word-level under a black-box setting. They applied BLEU as a score function to locate the victim words by measuring the difference between the original sentence and the sentence with target word replaced with a special token, and replaced these victim words with synonyms.

2.2 Attention in NMT

Attenion was first derived from human intuition based on the human activities, later adapted to machine translation for automatic token alignment (Hu, 2019). Attention mechanism, a simple method that can be used for encoding sequence data based on the importance score each element is assigned, has been widely applied to and attained significant improvement in various tasks in natural language processing, including sentiment classification, text summarization, question answering, dependency parsing, etc. In this section, we will introduce some related work about attention mechanism in NLP.

The traditional machine translation models (Kalchbrenner & Blunsom, 2013) are constructed by an encoder-decoder architecture, both of which are recurrent neural networks. An input sequence of source tokens is first fed into the encoder, with which the tokens will be transferred to the hidden representations, and then the decoder will utilize these hidden representations from the encoders as the initial input and output a sequence of dependent tokens. Such an encoder-decoder framework had achieved highest performance compared to purely statistical machine translation models. However, this architecture suffers from two serious drawbacks. First, RNN is forgetful, meaning that old information cleaned up after being propagated over multiple time steps. Second, there is no explicit word alignment during decoding and therefore focus is scattered across the entire sequence. To this end, the concept of attention was first introduced for an encoder-decoder structured NMT by Bahdanau (2015), and has become popular in the NMT community as an essential component of sequence-to-sequence models. Bahdanau provided such an attention mechanism to model word alignments between input and output sequence, which is an essential aspect of structured output tasks such as translation or text summarization. Based on Bahdanau's attention, Luong (2015) proposed two attention models, namely local and global, in context of machine translation tasks. The global attention model is similar to Bahdanau's attention while the local attention is computed with hidden states from the output of the encoder. Luong's attention achieved a better performance than Bahdanau's attention and provided a way of transparentizing the NMTs.

Recurrent architectures rely on sequential processing of input at the encoding step that results in computational inefficiency, as the processing cannot be parallelized (Vaswani et al., 2017). To address this, Vasiwani proposed Transformer architecture that eliminates sequential processing and recurrent connections. Specifically, transformer-based architectures, which are primarily used in modelling language understanding tasks, avoid recurrent structure in neural networks and instead trust entirely on self-attention mechanisms

to draw global dependencies between inputs and outputs. To be more specific, the transformer views the encoded representation of the input as a set of key-value pairs,(K, V), whose dimension equals input sequence length. For the decoder, the previous output is compressed into a query Q and the next output is produced by mapping this query and the set of keys and values. Referring to Bahdanau's and Luong's attention, the transformer adopts the scaled dot-product attention: the output is a weighted sum of the values, where the weight assigned to each value is determined by the dot-product of the query with all the keys:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{n}}\right)\mathbf{V}.$$

Transformer architecture achieved significant parallel processing, shorter training time, and higher accuracy for Machine Translation without any recurrent component. Besides, self-attention can provide correlations among the contextual words for NLP models, which we will utilize in our proposed algorithm.

2.3 BERT and its variations

BERT evolution has multiplied into diverse domains over time. Descendent of the Transformer architecture, BERT is a Bidirectional Encoder Representation which is trained with two unsupervised tasks: masked language model, and next sentence prediction. BERT models are heavily pre-trained on millions and billions of unannotated texts allowing us to fine-tune the model on custom tasks and with specific datasets through a transfer learning. Due to the superior model structure and large training data, BERT has performed many state-of-arts in many NLP tasks such as GLEU (Wang et al., 2018), SQuADv1.1 (Rajpurkar et al., 2016), SQuASv2.0 (Rajpurkar et al., 2018), SWAG (Zellers et al., 2018), etc. In addition to the performance in language understanding, BERT has also become a ground-breaking framework for many natural language processing tasks such as Sentimental analysis, sentence prediction, abstract summarization, question answering, natural language inference, and many more. BERT has various model configurations, BERT-Base the most basic model with 12 encoder layers and BERT-Large model with an additional number of layers.

Over time many new models have been inspired by the BERT architecture but are trained in different languages or optimized on domain-specific data sets. One of well-known BERT variants is RoBERTa (Liu et al., 2019), known as a Robustly Optimized BERT Pretraining Approach, which is developed to enhance the training phase. RoBERTa was developed by training the BERT model longer, on larger data of longer sequences and large mini-batches. By such a setting, RoBERTa obtained substantially improved results with some modifications of BERT hyper-parameters. Besides, RoBERTa does not make next sentence prediction (NSP) and make dynamic word masking.

A lite version of BERT (ALBERT) (Lan et al., 2020) was another well-known version of BERT. It was proposed to enhance the training and results of BERT architecture by using parameter sharing and factorizing techniques to reduce the number of parameters. BERT model contains millions of parameters, BERT-based holds about 110 million parameters which makes it hard to train also too many parameters impact the computation. To overcome such challenges ALBERT was introduced as it has fewer parameters compared to BERT.

3 Methodology

In this section, we first introduce and formulate the attention mechanism in NMT. Then, we elaborate on the proposed two-step attentive adversarial attack to NMTs, which features an attentive word location and a semantic-aware word substitution. Specifically, we firstly calculate the Hybrid Attention weights consisting of the language-specific translation attention and sequence-centered self-attention to locate the sensitive words. Then, we target to find replacement words using costume-designed selection steps to ensure parsing correctness and semantic preservations.

3.1 Attentions in NMT

Bahdanau (2015) proposed the attention mechanism to help the word alignments, especially for long sentences. We argue that such an attention mechanism reflects the contributions of each input words to the translated results, therefore a small perturbation to the most contributing word will give a heavy influence to the translation. The attention model utilizes a encoder-decoder framework for each step j during decoding they compute an attention score α_{ji} for hidden representation $\boldsymbol{h}$ in i of each input token to obtain, and the formulation is below:

$$e_{ji} = a\left(\boldsymbol{s}_i, \boldsymbol{h}_j\right) \tag{1}$$

$$\alpha_{ij} = \frac{\exp\left(e_{ij}\right)}{\sum_{k=1}^{T} \exp\left(e_{ik}\right)} \tag{2}$$

$$c_j = \sum_{j=1}^{T} \alpha_{ji} \boldsymbol{h}_i, \tag{3}$$

where e_{ji} is output of an alignment model a, usually a forward neural network, and $\boldsymbol{s_i}$ is the decoder RNN hidden state for time i. Using e_{ji}, one can score how well the inputs around position j and the output at position i match. c_{ji} is the encoded sentence representation with respect to the current element $\boldsymbol{h}_j$ to measure its similarity with output sequence (y_1,y_2...y_t), where y_1 is the t-th output tokens. The diagram for the this attetion model is demonstrated in Fig. 1.

Self-Attention (Vaswani et al., 2017) can be applied to many other kinds of NLP tasks besides machine translation. Different from a translation task, the goal is to learn the dependencies between the words in a given sentence and use that information to capture the internal structure of the sentence. In self-attention, there are 3 important variables, Q, K and V, which are vectors used to get better encoding for both our source and target words. All of these three variables are hidden representations from the linear layer. Futhurmore, the attention weights of self-attention is also calculated different with Bahdanau's attention, the formulation is below:

$$\text{Self} = \text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^{\top}}{\sqrt{(d_k)}}\right)\mathbf{V}. \tag{4}$$

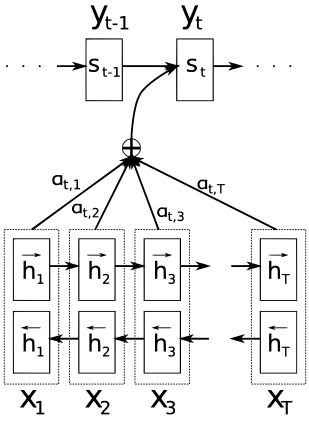

Fig. 1 Illustration of an attention-based NMT model (Dzmitry Bahdanau & Bengio, 2015) with RNN based encoder-decoder structures, generating the t-th target token y_t given a input sentence (x_1, x_2 ,..., x_T)

where d_k is the number of dimensions for key vector K. We argue to attack NMTs using self-attention too, as an disturbance to the dependency of source language can also deprave the translation quality.

3.2 Problem formulation

Denoting the source sequence as S, the translated target sequence as Y, a NMT model can be defined as $f(S) : S \rightarrow Y$. We denote $S = [w_1, \ldots, w_n]$ and $Y = [h_1, \ldots, h_k]$, where w and h denote the words in the source and target sequence, while n and k are the number of words in each respective sequence. To ensure the attack's applicability, we assume a black-box setting where the attacker can only query the NMT model for translated results of a given input, and does not have access to the model parameters, gradients or training data. For an input pair (S, Y), we want to generate an adversarial example S_{adv} such that $f(S_{adv})$ has an obvious semantic difference from Y. Additionally, we want S_{adv} to be grammatically correct and semantically similar to S.

3.3 Attentive word location

Attention weights in NMT models can be seen as the strength of semantic association between the source and target tokens, by adopting such a mechanism, the performance NMTs are boosted (Dzmitry Bahdanau & Bengio, 2015). Hence, we argue that NMTs can be crashed if the attention mechanism is tampered, and the best way of tampering attention is to adopt attention mechanism itself. In this subsection, we introduce the proposed attentive word location scheme and demonstrate different attentive NMT attack implementations based on language-specific and sequence-centered attentions.

3.3.1 Translation attentive attack

Since translation is a cross-language task defined by the source and target languages, it is intuitive to pose language-specific attacks to challenge NMTs' robustness. To this end, we propose a Translation Attentive Attack (TAA) mechanism that focuses on influential words in the translation towards a certain target language. Concretely, we obtain such an attention

$\mathcal{A}$ that measures word-wise importance in a specific translation task based on a contextual NMT model (Dzmitry Bahdanau & Bengio, 2015).

To calculate $\mathcal{A}$, we feed the NMT model with the source sequence to get the translated result $\hat{Y} = [\hat{h}_1, \dots, \hat{h}_{k'}]$, where k' is the number of words in the attacked target sentence. We then extract a correlation matrix $\mathcal{A}$ from the softmax layer in the model's decoder, thereby formulating the process as $\mathcal{T}(S) : S \to \mathcal{A}$. The elements in the correlation matrix $\mathcal{A}$ describe the probability distributions of translated words in the target language conditioned on the source sequence S, which can be written as:

$$a_{ij} = P(\hat{h}_j|[w_1, \dots, w_i, \dots]) = \frac{\exp\left(e_{ij}\right)}{\sum_{i=1}^{n} \exp\left(e_{ij}\right)}, \tag{5}$$

where P denotes probability, and e_{ij} denotes the feature in the model depicting the matching degree between the predicted word $\hat{h}_j$ in the target language and the input word w_i in S. The conditional probabilities reveal the correlation between the input sequence and the predicted sequence in the target language. Given its softmax-normalized distribution, we have $\sum_{i=1}^{n} a_{ij} = 1, \forall j$, therefore it is intuitive to measure w_i's contextual contribution to a translated word $\hat{h}_j$ using a_{ij} straightforwardly. Further, to find the most influential input words in the translation process, for the whole predicted sequence, we define the language-specific word-wise attention by summing the matrix elements by index j, as $\mathbb{A} = [A'_1, \dots, A'_i, \dots, A'_n]$, where $A'_i = \sum_{j=1}^{k'} a_{ij}$.

We can sort the words of the source sequence according to such an attention weight, $\mathbb{A}$, for the first step, and select the top language-specific influential words as the victim words for substitution in the second step, which will be introduced in Sect. 3.4.

3.3.2 Self-attentive attack

Beside the language-specific attack that focuses on the translation task between two languages above, the inherent semantics of the input sequence can also be tampered. Thus we propose a sequence-centered Self-Attentive Attack (SAA) which exploits attention from the input sequence itself. We utilize the transformer model (Vaswani et al., 2017), $\mathcal{V}(S) : S \to \mathcal{B}$, to extract the self-attention matrix $\mathcal{B}$, whose elements b_{ij} indicate the word-wise weights given positional encodings. Particularly, since such weights are obtained via softmax activation, they are also naturally normalized ($\sum_{i=1}^{n} b_{ij} = 1, \forall j$), and thus they are suitable to quantitatively measure the dependencies among words across the entire input sequence. Therefore, similar to the first step in TAA, we define the sequence-centered self-attention weight as $\mathbb{B} = [B'_1 \dots B'_i \dots B'_n]$, where $B'_i = \sum_{j=1}^{n} b_{ij}$.

Different from the language-specific attention in TAA that emphasizes on contextual alignment between source and target sequences, the sequence-centered attention in SAA can explore long-range dependencies within the input sequence itself, better indicating the word-wise influence on overall language understandings of the sequences.

3.3.3 Hybrid attentive attack

As analyzed above, the translation-attentive attack and self-attentive attack focus on different aspects of NMTs, i.e., the cross-language context alignment and the overall semantic understanding of the source sequence, respectively. We argue that both the two aspects are crucial

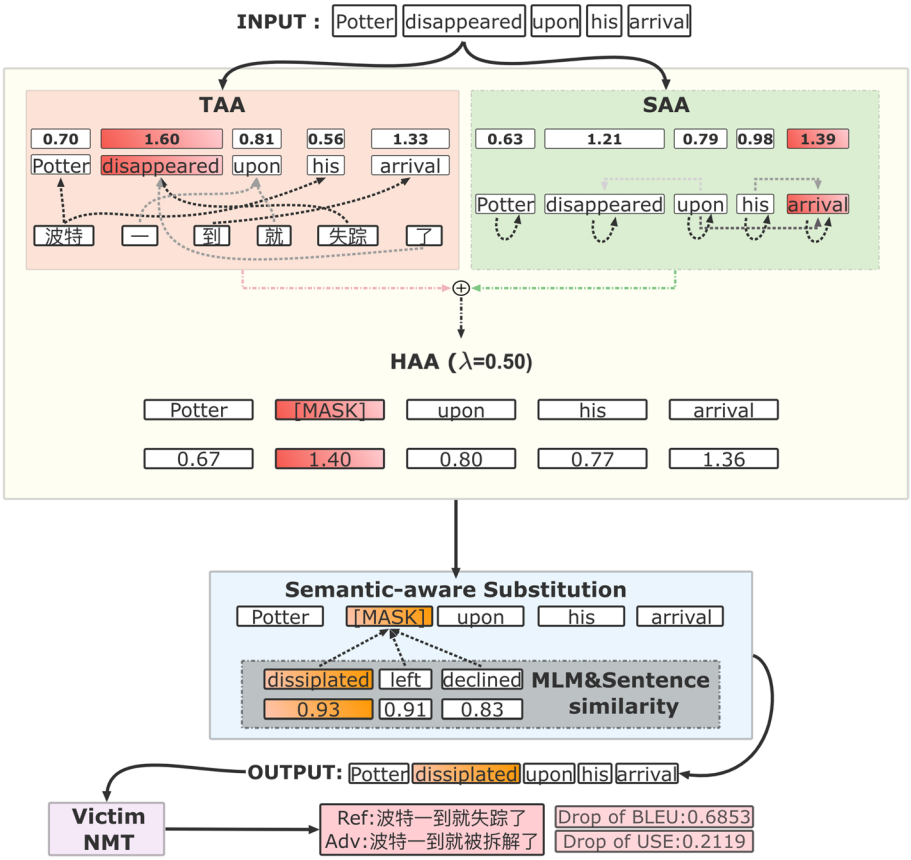

Fig. 2 An illustrated example of our HAA model. In this example, HAA generates an adversarial example with one word perturbed to attack an English-Chinese translation. The arrows inside the TAA box, and those in the SAA box, respectively represent the utilisation of translation and self-attention weights. The numbers inside the semantic-aware substitution box represents the sentence-level semantic similarity. The TAA, SAA, HAA and Semantic-aware Substitution workflows are reflected in lines 2–3, lines 4–5, line 6, and lines 7–15 in Algorithm 1, respectively

for NMTs, and an ideal attack for NMTs should combine their advantages. Thus we propose a Hybrid Attentive Attack (HAA) scheme which comprehensively considers the word influence by combining the attention weight from TAA and SAA:

$$\mathbb{H} = (1 - \lambda)\mathbb{A} + \lambda\mathbb{B}, \tag{6}$$

where $\mathbb{H} = [H'_1 \dots H'_i \dots H'_n]$ and H'_i is the final influence weight for word w_i in the input sentence. The optimal parameter λ can be found by a greedy search based on the attack performance measured by BLEU on translated results. The overall workflow of the HAA model is demonstrated in Algorithm 1 with an example shown in Fig. 2.

Algorithm 1 Hybrid Attentive Attack (HAA)

Input: Source and Target sentence pair S, Y, number of perturbed words N, number of adversarial candidates, N'

Model: $\mathcal{T}$: Translation attentive model for TAA. $\mathcal{V}$: Self-attentive transformer for SAA. $\mathcal{M}$: MLM model for word substitution.

Output: Adversarial Examples S_{adv}

Tokenize S
$\mathcal{A} \leftarrow \mathcal{T}(S)$ ▷ elements in $\mathcal{A}$ are represented by a_{ij}
$\mathbb{A} \leftarrow \sum_{j=1}^{n} a_{ij}$
$\mathcal{B} \leftarrow \mathcal{V}(S)$ ▷ elements in $\mathcal{B}$ are represented by b_{ij}
$\mathbb{B} \leftarrow \sum_{j=1}^{n} b_{ij}$
$\mathbb{H} \leftarrow (1-\lambda)\mathbb{A} + \lambda\mathbb{B}$
$S_{mask} \leftarrow$ mask the top N tokens by $\mathbb{H}$ scores
$\mathbb{S}_{can} = [\ ]$ ▷ create an empty set
for i in range(n^*) **do**
    $S'_{can} \leftarrow$ the ith highest replacement from $\mathcal{M}(S_{mask})$
    if $S'_{can} \neq S$ **then**
        $\mathbb{S}_{can}$.append(S'_{can})
    end if
end for
$S_{adv} \leftarrow$ the element of $\mathbb{S}_{can}$ that has the highest semantic similarity with S
$S_{adv} \leftarrow$ Detokenize S_{adv}
return S_{adv}

3.4 Semantic-aware word substitution

In the above subsection, we locate the most influential words in the input sequence to be attacked. An ideal attack should guarantee sufficient concealment besides having attack effectiveness, enabling the adversarial example to avoid being noticed by the NMT model. Therefore, we further argue an qualified adversarial example S_{adv} should preserve semantics and be grammatically correct, constraining reasonable deviations from the original input sequence.

We propose to design such a semantic-aware word substitution approach based on the semantic feature similarity between the tampered sequence and the original one. We mask a victim word one at a time by a descending order of the attention score to get S_{mask}, and utilise an MLM model $\mathcal{M}(S_{mask}) : S_{mask} \rightarrow S'_{can}$, where S'_{can} is a mask-filled sentence. At each iteration, we utilize $\mathcal{M}$ to generate n^* best adversarial example candidates, $\mathbb{S}_{can} = [S'_{(can,1)}, \ldots, S'_{(can,p)}, \ldots, S'_{(can,n^*)}]$, according to corresponding logits from $\mathcal{M}$, and we use a pre-trained semantic retrieval model, universal sentence encoder (USE) (Yang et al., 2019), to calculate the cosine feature distance between the candidate $S'_{(can,p)}$ and the original sequence S. Then we select S_{adv} with highest similarity to the original one as the adversarial example. By such a semantic-aware word substitution, we can complete the NMT adversarial attack process and strike a balance between influencing the translation result and concealing the perturbations with similar semantics.

4 Experiments

We empirically evaluated and assessed our proposed attacking strategies (TAA, SAA and HAA) on a task of translating English to Chinese to three well-performed world-leading NMTs: Google Cloud Translation, Baidu Cloud Translation and Helsinki NMT

Table 1 Introduces details about datasets used in the experiments

Dateset	YYeTs Subs	Commentary	Infopankki	Openoffice	WMT20 T1	WMT20 T2	ALT.P (test)
Size	500k	69k	30k	69k	6.0k	6.0k	1.0k
Avg.len	7.83	46.14	9.92	6.16	14.10	16.51	16.54
Min.len	1	1	1	1	3	2	2
Max.len	67	229	144	221	130	199	204
Content	Movie subs	News	Science	Education	Wikipedia	Wikipedia	News
Purpose	Training Set				Testing set		

(Tiedemann, 2020). To deeply explore the attacking performance, we not only attack the victim model but also make transfer attacks which utilize the adversarial examples generated on one victim NMT to attack other NMTs.

4.1 Datasets

To get sufficient training data, we utilized 4 datasets as our training set for training the language-specific NMT and sequence-centered transformer models utilized for the TAA, the SAA, and the MLM for semantic-aware word substitution. Three of the training sets are Commentary (Tiedemann, 2012), Infopankki (Tiedemann, 2020) and the Openoffice (Tiedemann, 2020), are publicly available, while the other, YYeTs subs,[1] is scripted by us from YYeTs website (provided in the supplementary material), which provides human translated movie and drama subtitles. The details of the train set can be found in Table 1.

To get reliable experimental results, we test attacking strategies on 3 other public datasets, WMT20 T1, WMT20 T2 (Tiedemann, 2020) and ALT-P(test) (Riza et al., 2016). WMT is the main event for machine translation and machine translation research, which provides reliable multilingual datasets from Wikipedia. To diverse the sources of test set, we also include ALT-P dataset on news. The details of the test set can be found in Table 1.

4.2 Victim models

We test the proposed attacking strategies on three well-performed NMTs: Google Cloud Translation[2] (Google.T), Baidu Cloud Translation[3] (Baidu.T), and Helsinki NMT (Hel.T) (Tiedemann, 2020). The first two NMTs are cloud translation platforms, which are used for commercial purposes while the other NMT, Helsinki NMT is based on MarianNMT(Junczys-Dowmunt et al., 2018) from Microsoft for academic purpose.

4.3 Baselines

We compare our proposed strategies with 5 word-level attack strategies below:

[1] https://m.yysub.net/
[2] https://cloud.google.com/translate
[3] https://api.fanyi.baidu.com/

- RAND: randomly selects victim words in the target sentences and utilize the proposed semantic-aware substitution strategy to construct the adversarial examples.
- Morpheus-Attack (Morph) (Tan et al., 2020), greedily searches for words, from *noun*, *verb*, or *adjective* tags, maximally decreasing BLEU on source language side, and substitute them with synonyms.
- BERT-ATTACK (BERT.A) (Li et al., 2020): utilizes BERT to locate the victim words by ranking the differences between the logits of original words and BERT-predicted words, and then make substitutions with BERT.
- Seq2sick (Cheng et al., 2020): crafts the adversarial example by depraving the targeted logits of victim NMT with regularization on preserving semantic similarity.
- PSO (Zang et al., 2020): selects word candidates from HowNet and employs the PSO to find adversarial text for classifier. We adjust the metric from classification logits to BLEU.

4.4 Evaluation metrics

We use metrics based on BLEU and USE (Yang et al., 2019) to evaluate attacking performance on the target language side and the semantic preservation on the source language side. BLEU evaluates the sentence pairs in term of word alignment while USE is a multilingual pre-trained language model to evaluate the semantic similarity.

Since changes of the original input will always lead to changes of the translated output, we examine how much more changes an attacked output has compared to those of the unattacked translation. So instead of directly using BLEU and USE on translated outputs, we define BLEU drop ratio (BDR) and USE drop ratio (UDR) to evaluate attacks:

$$\text{BDR} = \frac{\text{BLEU}(Y, f(S)) - \text{BLEU}(Y, f(S_{adv}))}{\text{BLEU}(Y, f(S))} \tag{7}$$

$$\text{UDR} = \frac{\text{USE}(Y, f(S)) - \text{USE}(Y, f(S_{adv}))}{\text{USE}(Y, f(S))} \tag{8}$$

where S and Y denote input sentence and translation reference, and $f(\cdot)$ is the victim NMT model.

In addition, we also evaluate how much word perturbations are made on the original inputs by using BLEU and USE on the attacked source language. To distinguish from the metrics used on the target language side, we use S-BLEU and S-USE for denoting changes made on the source language.

4.5 Experimental settings

In this section, we will introduced the models used for HAA and results of greedy searching λ. Since the number of of attacked words could intuitively affect the attacking performance, the number of perturbed words in each sentence to be attacked ranges from 1 to 5 in our experiment comparisons.

4.5.1 Model structures

In this subsection, we introduce the structure of the language-specific NMT for TAA, transformer for SAA, and the MLM for semantic-aware word substitution. All of these 3 models are trained and fine-tuned on the same train datasets mentioned in Table 1.

- *TAA*: The architecture of TAA consists of a 2-layer stacked LSTM, plus a Luong's (2015) translation attention layer to process of the output of LSTM. To be more specific, the encoder takes a list of subtoken IDs to an embedding vector for each subtoken via an embedding layer. Further, we processes the embeddings into a new sequence with a LSTM. After encoding, the features of input sentences will be passed into a decoder, and the decoder's job is to generate predictions for the next output token. The decoder receives the complete encoder output and uses a LSTM to keep track of what it has generated so far. To get translation attention, the decoder will utilize its LSTM output as the query to the attention over the encoder's output, producing the context vector. After the LSMT in decoder, we adopt the Luong's translation attention to combine the LSTM output and the context vector generate the translation attention matrix. For the last step, decoders generates logit predictions for the next tokens based on the attention matrix. For the hyper-parameter, we set 1024 hidden units, 256 embedding dimmensions, 64 batch size, with Adam optimizer.
- *SAA*: SAA is designed to get sequence-centered attention weights on the source language, therefore it will be trained with only the data in source language. Since the data is unlabeled and sequential, we utilize BERT-base-uncased (Devlin et al., 2019), one of the best unsupervised language models, as the transformer to extract the sequence-centered attention weights. The hyper-parameters of this model are public available. To adjust the model to our dataset, it will be fine-tuned on our datset with Adam optimizer with learning rate 0.001 and batch size 128.
- *MLM for semantic-aware substitution*: MLMs mask the words in the train set and are given a task to fill these masks, therefore utilize these models can help to find parsing substitutions for the proposed methods. We utilize a public pre-trained model, RoBERTa-large (Liu et al., 2019), as our candidate to generate parsing and semantic-preserving adversarial examples.

4.5.2 Optimization of λ

In the experiments, our proposed method, HAA, utilizes a greedy search for the best hyper-parameter λ to combine language-specific and sequence-centered attention. The objective used for searching is BLEU and the search is within the validation set which contains 1000 samples separated from the training set. We greedily search for the optimal hyper-parameter λ within $[0, 0.01, \ldots, 1]$ with a step size of 0.01 for each victim model and the searched results for the three victim NMTs (Google, Baidu and Helsinki translations) are shown in Fig. 3.

From searched results in Fig. 3, we can find that the optima λ values for the three victim models are $\lambda_{\text{Google}} = 0.68$, $\lambda_{\text{Baidu}} = 0.47$ and $\lambda_{\text{Hel.T}} = 0.41$. Therefore, we can find the λ can be different for different victim NMTs in our experimental settings. Since λ is utilized to control the weight of SAA and TAA, it can show the preference between SAA and TAA. From the results, we find that for different victim NMTs, the proposed

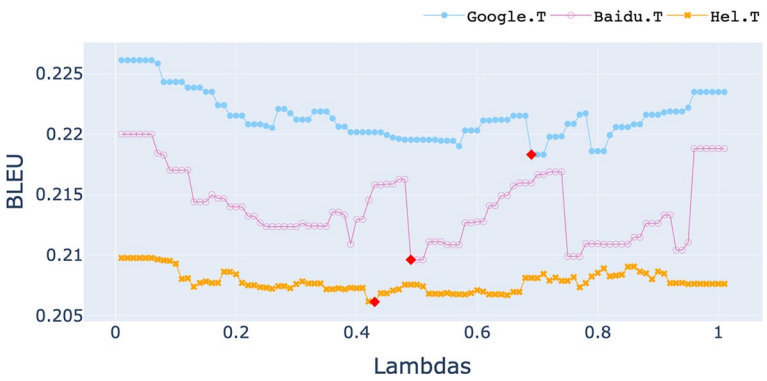

Fig. 3 The process of searching for the best λ for Google, Baidu and Helsinki NMT. The discovered optimal λ values are highlighted in red (Color figure online)

HAA will have different preferences: TAA is preferred for Google translation while SAA is preferred for Baidu Translation and Helsinki Translation. Besides, as the λ is searched based on the performance of NMTs, there is no doubt that the λ can be different due to the different NMTs' performance on datasets so that this preference can be different in datasets.

4.6 Main results and analysis

We show the results for greedy searching process in Fig. 3. The main results of attacking performance and semantic preserving performance on different test data sets are shown in Tables 2, 3, 4, and Figs. 4, 5, and 6. In addition to the statistics of the results, an example of learned attentions for the proposed methods is shown in Table 5 and an adversarial example is also shown in Table 6 to show the differences of attacks. We validate the advantages of our proposed methods (i.e., TAA, SAA and HAA) from the following three aspects:

4.6.1 Does HAA have superior attack performance compared to baselines?

We compare the attacking performance of the proposed attentive methods (TAA, SAA, and HAA) and non-attentive baselines in Fig. 2, reflected by decreases of BLEU and USE between the original and the attacked translation results. It can thus be concluded that the proposed method HAA achieves the best attacking performance, with the largest metric score drops for both word alignment (BLEU) and semantic understanding (USE). Particularly, as shown in Fig. 2, HAA consistently outperforms other competing methods across different data domains, regardless of the number of perturbed words. Apart from HAA itself, its different attentive components TAA and SAA also show surpass the non-attentive baselines in most cases.

Table 2 Comparisons of performance on WMT20 T1 dataset in terms of semantic preservation (S-BLEU, S-USE) and attacking performance (BDR, UDR) averaged across different number of perturbed words for victim models

Attack method	Google.T				Baidu.T				Hel.T			
	Avg. BDR (%)	Avg. UDR (%)	Avg. S-BLEU	Avg. S-USE	Avg. BDR (%)	Avg. UDR (%)	Avg. S-BLEU	Avg. S-USE	Avg. BDR (%)	Avg. UDR (%)	Avg. S-BLEU	Avg. S-USE
RAND	15.94	4.53	0.3637	0.8124	19.81	7.26	0.3600	0.8243	14.22	5.87	*0.3634*	0.8225
BERT.A	22.20	9.55	0.3622	0.7431	22.90	10.85	0.3611	0.7723	19.20	7.77	0.3602	0.7712
PSO	23.81	7.65	**0.3659**	0.8312	24.80	12.42	0.3671	**0.8325**	18.37	10.11	0.3605	0.8321
Morph	32.69	7.65	0.3601	*0.8319*	34.78	15.78	0.3618	0.8303	25.04	11.84	0.3637	0.8301
Seq2sick	27.97	9.34	0.3552	0.7508	37.69	15.88	0.3621	0.7590	20.86	13.11	0.3631	0.7545
SAA	38.48	16.59	0.3631	0.8312	*46.94*	28.04	*0.3646*	0.8305	34.19	14.86	*0.3634*	**0.8342**
TAA	*36.87*	*15.92*	0.3639	0.8377	47.75	*22.09*	0.3641	*0.8310*	*32.51*	*14.19*	**0.3647**	*0.8333*
HAA	**47.17**	**22.72**	0.3639	**0.8395**	**54.59**	**25.67**	**0.3794**	0.8319	**47.66**	**20.03**	0.3632	0.8334

For each column, the highest, the second and third highest score are highlighted in bold, underlined and italic, respectively

Table 3 Comparisons of performance on WMT20 T2 dataset in terms of semantic preservation (S-BLEU, S-USE) and attacking performance (BDR, UDR) averaged across different number of perturbed words for victim models

Attack method	Google.T				Baidu.T				Hel.T			
	Avg. BDR (%)	Avg. UDR (%)	Avg. S-BLEU	Avg. S-USE	Avg. BDR (%)	Avg. UDR (%)	Avg. S-BLEU	Avg. S-USE	Avg. BDR (%)	Avg. UDR (%)	Avg. S-BLEU	Avg. S-USE
RAND	20.71	4.86	0.3609	*0.8288*	22.31	9.63	0.3619	0.8288	21.18	7.59	0.3622	0.8288
BERT.A	25.98	10.79	0.3601	0.7660	32.07	15.04	0.3621	0.7677	26.18	13.32	0.3634	0.7632
PSO	25.92	9.35	**0.3666**	**0.8400**	32.60	14.87	0.3671	**0.8388**	26.63	12.13	**0.3655**	**0.8445**
Morph	32.77	9.14	0.3611	0.8339	38.77	17.11	0.3617	*0.8321*	31.05	14.22	0.3647	0.8308
Seq2sick	29.69	14.70	0.3599	0.7538	40.13	16.66	0.3633	0.7598	26.58	15.88	*0.3643*	0.7548
SAA	38.58	*17.10*	*0.3643*	0.8278	*45.38*	*20.19*	*0.3649*	0.8319	36.02	18.79	0.3639	0.8349
TAA	*35.03*	17.78	0.3645	0.8265	48.27	21.26	0.3647	0.8311	*35.05*	*18.92*	0.3641	0.8322
HAA	**44.12**	**20.53**	0.3633	0.8275	**53.85**	**25.67**	**0.3677**	0.8332	**41.64**	**23.58**	0.3642	*0.8328*

For each column, the highest, the second and third highest score are highlighted in bold, underlined and italic, respectively

Table 4 Comparisons of performance on ALT.P dataset in terms of semantic preservation (S-BLEU, S-USE) and attacking performance (BDR, UDR) averaged across different number of perturbed words for victim models

Attack method	Google.T				Baidu.T				Hel.T			
	Avg. BDR (%)	Avg. UDR (%)	Avg. S-BLEU	Avg. S-USE	Avg. BDR (%)	Avg. UDR (%)	Avg. S-BLEU	Avg. S-USE	Avg. BDR (%)	Avg. UDR (%)	Avg. S-BLEU	Avg. S-USE
RAND	15.34	18.04	0.3312	**0.7932**	10.00	15.88	0.3410	**0.7499**	19.71	12.05	0.3297	0.7361
BERT.A	17.28	21.17	0.3301	0.7362	30.015	18.16	0.3329	0.7477	23.34	14.90	0.3333	*0.7332*
PSO	20.05	25.00	0.3262	0.7410	23.15	16.69	0.3252	0.7388	26.71	15.74	0.3201	0.7155
Morph	28.36	26.35	0.3301	0.7321	34.04	21.07	*0.3371*	0.7332	28.10	17.55	*0.3331*	0.7201
Seq2sick	20.05	25.01	0.3211	0.7191	33.07	20.16	0.3231	0.7214	32.89	20.07	0.3243	0.7011
SAA	*40.63*	*30.20*	**0.3643**	0.7319	*46.01*	25.13	0.3321	0.7329	*35.18*	*22.72*	**0.3400**	0.7321
TAA	41.17	32.09	*0.3334*	0.7293	51.06	25.13	0.3347	0.7391	37.66	26.43	0.3234	0.7310
HAA	**46.51**	**35.59**	0.3353	*0.7377*	**57.42**	**31.01**	**0.3417**	*0.7410*	**48.34**	**34.30**	0.3299	**0.7388**

For each column, the highest, the second and third highest score are highlighted in bold, underlined and italic, respectively

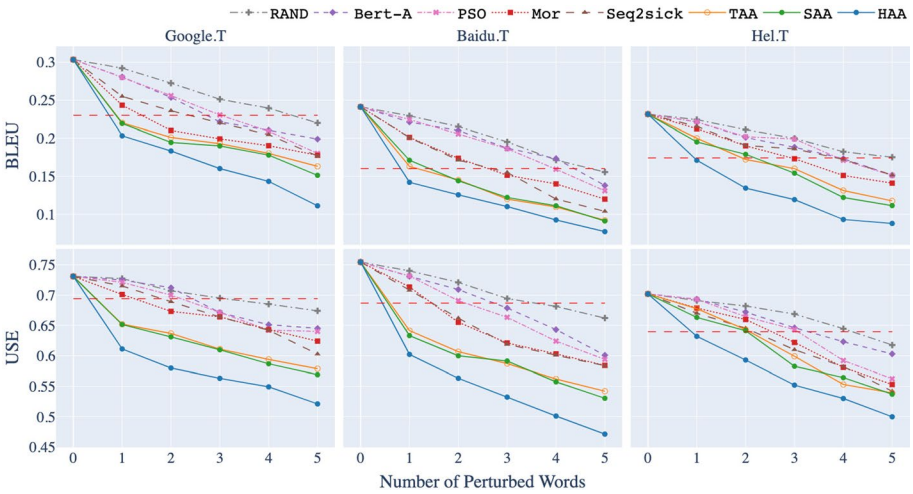

Fig. 4 Attacking performance (BLEU, USE) on the WMT20 T1 dataset towards different numbers of perturbed words ranging from 1 to 5 for three victims, NMT, Goolge.T, Baidu.T and Helsinki.T

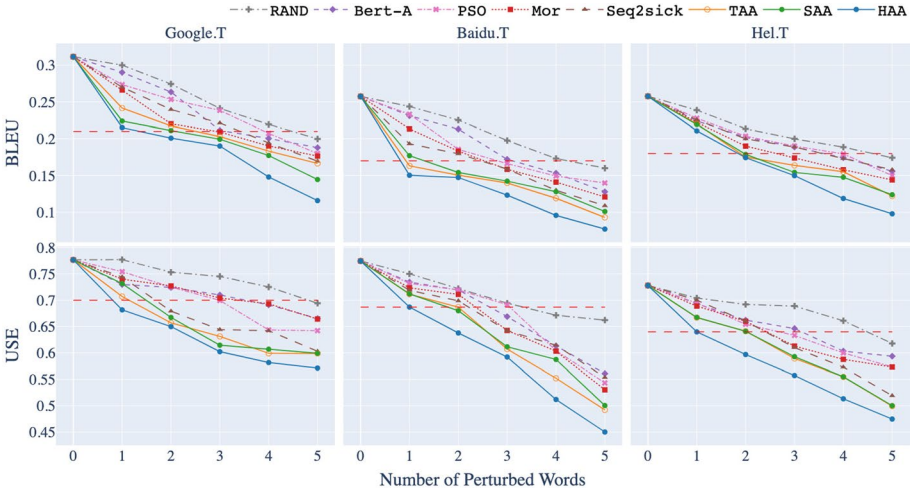

Fig. 5 Attacking performance (BLEU, USE) to the WMT20 T2 dataset towards different numbers of perturbed words ranging from 1 to 5 for three victims, NMT, Goolge.T, Baidu.T and Helsinki.T

4.6.2 Balance between attack performance and the number of perturbed words

Concerning the trade-off between effectiveness and imperceptibility, we evaluate the attack's imperceptibility from both appearance and semantic modification perspectives, the first of which is the number of words perturbed. As shown in Figs. 4, fig:T2 and fig:alt, comparing the numbers of words needed to achieve identical drops of metric scores (marked by the horizontal red dashed lines), we can find that HAA perturbs the fewest words, for it theoretical focuses on the most influential words with both language-specific

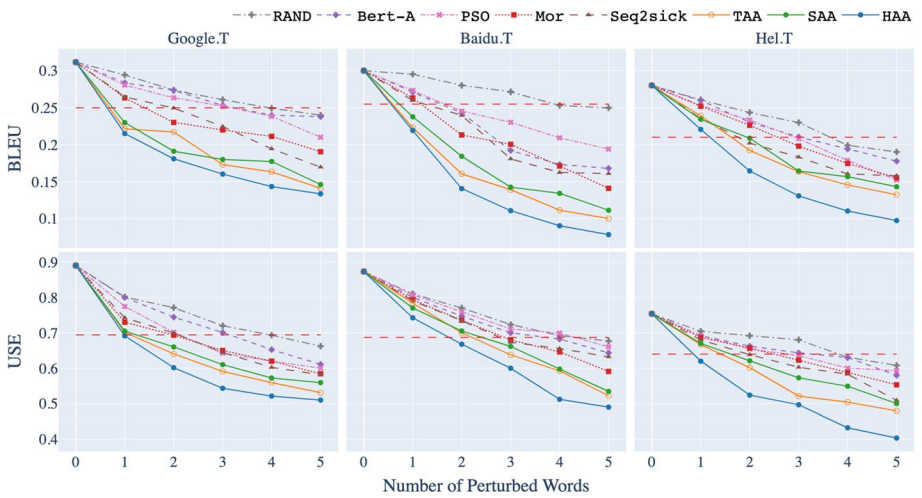

Fig. 6 Attacking performance (BLEU, USE) to the ALT.P dataset towards different numbers of perturbed words ranging from 1 to 5 for three victims, NMT, Goolge.T, Baidu.T and Helsinki.T. The horizontal red dashed lines indicate the numbers of words needed to achieve identical drops of metric scores

and sequence-centered attentions. Thus we can conclude that the proposed HAA more successfully balances attacking performance and the appearance modifications to the sequence.

4.6.3 How well does HAA reserve the semantic meaning of the original input sentences?

To further investigate the attack's imperceptibility, we evaluate the semantic similarities between the original input sentence and its derived adversarial sample (i.e., S-BLEU and S-USE) shown in Tables 2, 3 and 4 on different datasets. All of the table demonstrates the attacking methods based our semantic-aware substitution, SAA TAA HAA and RAND, are the best methods in most cases in terms of semantic preserving. In some cases, our methods are not the best, but they are still comparable to the best method PSO by a close margin in semantic preservation. However, PSO's preservation comes at the price of much inferior performance, as is shown by its BDR and UDR. Thus we can conclude that proposed HAA provides the one of the best balances between attack performance and semantics preservation.

To further validate the effectiveness of our word replacement strategy, we conduct an extensive experiment on our semantic-preserving performance by a task of substituting the same victim words located by our hybrid attention. We select 3 common substituting baselines:

- Default masked-word filling (HA.Def): utilize MLM to fill the mask without a consideration to the semantic preservation
- Synonyms (HA.Syn): replace the victim words with synonym from the WordNet (Miller, 1998)

Table 5 Examples for attentions learned by proposed methods (TAA, SAA and HAA). The examples are red, blue and green for TAA, SAA, and HAA, respectively

TAA	However, [0.68] after [1.58] a [3.26] few [4.16] victories, [6.68] the [6.40] campaign [9.37] falters. [7.74]
SAA	However, [0.80] after [0.79] a [0.68] few [0.98] victories, [0.75] the [0.72] campaign [0.79] falters. [0.94]
HAA	However, [0.75] after [1.10] a [1.72] few [2.25] victories, [3.12] the [2.99] campaign [4.22] falters. [3.66]

The opacity of each word depends on its corresponding attention weight which is placed in the brackets after each token

Table 6 Adversarial examples (adv.) crafted by proposed methods and baselines, and their corresponding translated results (Tran.)

BERT.A	Adv. This atrocity sparked off the war of the mustache, a millennia long that saw the empires of the Elves and Dwarves crumble into ruins (S-BLEU: 0.9440, S-USE: 0.9865)
	Tran. 这场暴行引发了胡子战争, 这场长达数千年的冲突中,精灵帝国和矮人帝国陷入废墟。(BDR: 3.31%, UDR:+3%)
Morph	Adv. This atrocity sparked off the war of the mustache, a millennia spanning that saw the empires of the Elves and Dwarves disintegrate into ruins. (S-BLEU: 0.9120, S-USE: 0.9193)
	Tran. 这一暴行引发了胡子战争,在长达数千年的战争中,精灵帝国和矮人帝国解体。(BDR:+7.76%, UDR: 5.19%)
PSO	Adv. This atrocity sparked off the war of the mustache, one millennia spanning that saw the empires of the Elves and Dwarves crumble into ruins. (S-BLEU: 0.9669, S-USE: 0.9958)
	Tran. 这场暴行引发了胡子战争,长达数千年 之久,精灵 和矮人帝国 崩溃。 (BDR: 23.37%, UDR: 1.90%)
Seq2sick	Adv. This atrocity sparked off the war of the mustache, a millennia spanning that saw the king of the Elves and Dwarves crumble into ruins. (S-BLEU: 0.9460, S-USE: 0.9663)
	Tran. 这场暴行引发了胡子战争,长达数千年 之久,精灵 和矮人 国王们 都陷入了困境。 (BDR: 21.33%, UDR: 5.789%)
SAA	Adv. This atrocity sparked off the war of the mustache, a millennia spanning that saw the empires of the Elves and Dwarfs crumble into ruins. (S-BLEU: 0.9739, S-USE: 0.9832)
	Tran. 这场暴行引发了胡子战争,长达数千年的 中,精灵帝国和矮人帝国 奔溃。 (BDR: 21.33%, UDR: 5.78%)
TAA	Adv. This atrocity sparked off the war of the beard, a millennia spanning conflict that saw the empires of the Elves and Dwarves crumble into ruins. (S-BLEU: 0.9329, S-USE: 0.9420)
	Tran. 这场暴行 是胡子战争引发的, 几千年来, 精灵 和矮人帝国 都陷入困境。 (BDR: 37.43%, UDR: 13.86%)
HAA	Adv. This atrocity sparked off the war of the beard, a millennia spanning conflict that saw the empires of the Elves and Dwarves crumble into ruins. (S-BLEU: 0.9329, S-USE: 0.9420)
	Tran. 这场暴行 是胡子战争引发的,几千年来,精灵 和矮人帝国 都陷入困境。 (BDR: 37.43%, UDR: 13.86%)

The semantic preserving (S-BLEU, S-USE) and attacking performance (BDR, UDR) metrics are provided in the brackets after the adversarial and translated sentence, respectively. The translation attacked by HAA made a completely wrong causality of between the "war" and the "mustache" by stating "The war of beard sparked off this atrocity" in the translation

– Word embedding distance ranking (HA.Rank): search the word embedding space in GloVe (Pennington et al., 2014) to set the word, with smallest distance (l_2) to victim word, as the replacement.

Table 7 Comparisons among different word substituting methods

Metrics	HA.Def	HA.Syn	HA.Rank	HAA
Avg.S-BLEU	0.3634	0.3521	0.3631	0.3642
Avg.S-USE	0.7639	0.6733	0.7591	0.8328

The results from Table 7 show that HAA (semantic-aware substitution) achieves the best semantic-preserving performance on attacking the same position. Clearly, HAA can provide more parsing-correct and semantic-preserved adversarial examples than other methods.

4.7 Transferability

The transferability of adversarial examples is defined as whether the adversarial examples targeting at a specific model f can also mislead another model f'. To evaluate transferability, we apply one-word-perturbation adversarial examples generated by different methods on mBART-large-cc25 (Tang et al., 2020), a sequence-to-sequence transformer from Facebook, to attack Googl e, Baidu and Helsinki translation models. Figure 7 shows the results on the original mBART NMT and other transferred models. It can be concluded from this figure that our attentive methods (TAA, SAA, and HAA) achieve the best attack performance on the three transferred NMT models, demonstrating the effectiveness of our methods in terms of attack transferability.

4.8 Attacking preference

As the superiority of proposed method in terms of attacking performance, we collect some statistics to research the attacking preference, described by speech (POS) tags, for different attacking strategies. In this subsection, we analyze statistics on POS as shown in Table 8, and aim to analyse the more vulnerable POS tags by a comparison between the proposed methods and baselines.

Words that are assigned to the same part of speech (POS) tags generally present similar syntactic importance, we investigate attacking strategies' preference on POS tags for further lingual analysis. We apply Stanford PSO tagger (Toutanova et al., 2003) to annotate them with POS tags, including *noun*, *verb*, *adjective (Adj.)*, *adverb (Adv.)* and *others* (i.e., pronoun preposition, conjunction, etc). Statistical results in Table 8 demonstrate that generally all the attacking methods tend to focus on *noun*, which we can suppose is the most sensitive POS category for translation. However, the proposed attacking strategies (TAA, SAA and HAA) tends to take a larger proportion of *Verbs* than any other methods, thus we may conclude that *Verb* might be the second adversarially vulnerable POS tag.

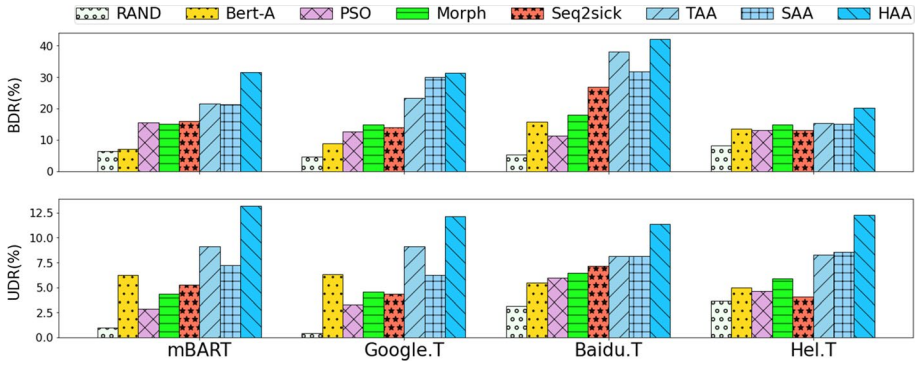

Fig. 7 Attacking performance (BDR, UDR) of transferred attacks from mBART to Google, Baidu and Helsinki NMT models

Table 8 Distributions of POS tags for different attack strategies The percentages are calculated row-wise.

Models	Noun	Verb (%)	Adj. (%)	Adv. (%)	Others (%)
PSO	**40.89**	9.12	15.77	17.89	*16.33*
Seq2sick	**40.11**	14.90	19.30	8.77	*16.92*
BERT.A	**78.42**	3.90	9.92	2.51	*5.25*
Morph	35.51	9.77	**44.19**	*10.53*	0.0
TAA	**48.37**	24.33	6.15	0.04	*17.15*
SAA	**44.71**	27.10	*17.56*	6.57	4.06
HAA	**51.14**	23.86	*17.41*	2.79	4.80

For each row, the most, second and third highest percentage is highlighted in bond, underlined italic, respectively

5 Discussion and Conclusions

In recent years, safety and fairness of NLP models have greatly been threat by adversarial attacks. Most existing researches focus on NLP classifiers, such as fake news detection, sentiment analysis, and email spam while few researchers raise concern about the robustness of the sequence-to-sequence neural machine translation (NMT) models. Unlike the classifiers, NMT outputs a sequence of dependent discrete classes or token IDs rather a single class. To this end, the attacking performance for NMTs would perform poorly if the victim words only affect their translation results while the semantics of other words are still preserved. Thus, to make a threat level attacks to NMTs, the attackers should not perturb the victim words only but also the contextual environment.

In this research, we have proposed HAA which selects influential words by both translation-specific and language-centered attentions and substitutes them with semantics preserved word perturbations. Adversarial examples generated by our proposed method will not only affect the victim words translation but also other words' translations. Experiments demonstrate that HAA delivers the best balance between the number of perturbed words and attacking performance among the competing methods.

Although the generated adversarial examples can threat the NMTs, adversarial examples are not bugs but features (Ilyas et al., 2019). To protect the NMT from the proposed attack, we believe that one possible defence strategy is adversarial retraining, which is usually done by joining the adversarial examples in the training set then retraining the models with the newly constructed training set. Although we did not perform the adversarial retraining in experiments, due to the lack of access to the victim models' structure since the Google and Baidu translations are online service and Helsinki NLP does not specify their model structures, by joining the adversarial features into model training, the model can be theoretically more robust against adversarial attacks.

Since the adversarial attack is one of the most effective methods to test the robustness of a model, the proposed attentive attacks raise some concern about the attention mechanism. As transformers with attention mechanism achieved great success, most of the existing well-performed NLP models are based on such an mechanism. Such an popularity of attentions could put NMTs in high risks because attackers can make effective attacks by utilizing the attention mechanism. Thus a safer way of applying attentions is a promising future research direction. At the same time, we also plan to pertinently study and design defence strategies to further improve the robustness of NMT models under future adversarial attacks.

Author contributions MN contributed to conceptualization, theoretical analysis, experiments and draft preparation; CW contributed to experiments, draft preparation, writing-review, editing; TZ worked on theoretical-review, editing and searching resources; SY worked on theoretical-review, editing and searching resources; WL made contributions to conceptualization, theoretical analysis, and draft writing.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions. Not applicable.

Availability of data and materials All of the datasets are available on Huggingface (https://huggingface.co/datasets) and on our GitHub site (https://github.com/MingzeLucasNi/HAA.git).

Declarations

Conflict of interest Not applicable.

Ethics approval Not applicable.

Consent to participate The authors give their consent to participate.

Consent for publication The authors give their consent to the publication of all information in this paper.

Code availability All codes from our experiments are available at https://github.com/MingzeLucasNi/HAA.git

References

Alzantot, M., Sharma, Y., Elgohary, A., Ho, B.-J., Srivastava, M., & Chang, K.-W. (2018). Generating natural language adversarial examples. In *Proceedings of the 2018 conference on EMNLP*.

Belinkov, Y., & Bisk, Y. (2017).Synthetic and natural noise both break neural machine translation. In *International conference on learning representations (ICLR)*.

Berant, J., Chou, A. K., Frostig, R., & Liang, P. (2013). Semantic parsing on freebase from question-answer pairs. In *Emnlp*.

Cheng, M., Yi, J., Chen, P.-Y., Zhang, H., & Hsieh, C.-J. (2020). Seq2sick: Evaluating the robustness of sequence-to-sequence models with adversarial examples. In *Proceedings of the aaai conference on artificial intelligence*(Vol.34, pp. 3601–3608).

Cheng, Y., Jiang, L., & Macherey, W.(2019). Robust neural machine translation with doubly adversarial inputs. In *Proceedings of the 57th annual meeting of the association for computational linguistics* pp. 4324–4333.

Chivukula, A, S., & Liu, W.(2017). Adversarial learning games with deep learning models. In *2017 international joint conference on neural networks (ijcnn)* pp. 2758–2767.

Chivukula, A. S., & Liu, W. (2018). Adversarial deep learning models with multiple adversaries. *IEEE Transactions on Knowledge and Data Engineering, 316*, 1066–1079.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). Bert: pre-training of deep bidirectional transformers for language understanding. In *Naacl.*

Dzmitry Bahdanau, K. C., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. arXiv:1409.0473

Ebrahimi, J., Lowd, D., & Dou, D. (2018). On adversarial examples for character-level neural machine translation. In*Proceedings of the 27th international conference on computational linguistics*(pp. 653–663).

Gao, J., Lanchantin, J., Soffa, M., & Qi, Y. (2018). Black-box generation of adversarial text sequences to evade deep learning classifiers. *2018 IEEE Security and Privacy Workshops (SPW)*,50–56.

Goodfellow, I., Shlens, J., & Szegedy, C.(2015). Explaining and harnessing adversarial examples. *CoRR*, arXiv:1412.6572

Hu, D. (2019). An introductory survey on attention mechanisms in nlp problems. In *Proceedings of sai intelligent systems conference* pp. 432–448.

Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., & Madry, A.(2019). Adversarial examples are not bugs, they are features. In *Neurips*.

Junczys-Dowmunt, M., Grundkiewicz, R., Dwojak, T., Hoang, H., Heafield, K., Neckermann, T., Seide, F., Germann, U., Aji, A.F., Bogoychev, N., Martins, A.F., Birch, A. (2018). Marian: Fast neural machine translation in C++. In *Proceedings of acl 2018, system demonstrations*(pp. 116–121). Melbourne, Australia.

Kalchbrenner, N., & Blunsom, P. (2013).Recurrent continuous translation models. In *Proceedings of the 2013 conference on empirical methods in natural language processing*(pp. 1700–1709). Seattle, Washington, USA.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2020). Albert: A lite bert for self-supervised learning of language representations. arXiv:1909.11942

Li, L., Ma, R., Guo, Q., Xue, X., & Qiu, X.(2020). Bert-attack: Adversarial attack against bert using bert. In *Proceedings of the 2020 conference on empirical methods in natural language processing (emnlp)* pp. 6193–6202.

Li, Z., Wang, X., Li, J., & Zhang, Q. (2021). Deep attributed network representation learning of complex coupling and interaction. *Knowledge-Based Systems, 212*, 106618.

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

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy O, Lewis M, Zettlemoyer, L., Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. arXiv:1907.11692.

Luong, M., Brevdo, E., & Zhao, R. (2017). Neural machine translation (seq2seq) tutorial. https://github.com/tensorflow/nmt.

Luong, T., Pham, H., & Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. In *Emnlp*.

Michel, P., Li, X., Neubig, G., & Pino, J. (2019). On evaluation of adversarial perturbations for sequence-to-sequence models. In *Proceedings of the 2019 conference of the north american chapter of the association for computational linguistics: Human language technologies, volume 1*.

Miller, G. A. (1998). *Wordnet: An electronic lexical database*. MIT press.
Papernot, N., Mcdaniel, P., Swami, A., & Harang, R. E. (2016). Crafting adversarial input sequences for recurrent neural networks. *MILCOM 2016: 2016 IEEE Military Communications Conference* 49–54.
Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (emnlp)* pp. 1532–1543.
Rajpurkar, P., Jia, R., & Liang, P. (2018). Know what you don't know: Unanswerable questions for squad. In *Proceedings of the 56th annual meeting of the association for computational linguistics (volume 2: Short papers)* pp. 784–789.
Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 conference on empirical methods in natural language processing* pp. 2383–2392.
Riza, H., Purwoadi, M., Gunarso, Uliniansyah, T., Ti, A. A., Aljunied, S. M., Mai, L.C., Thang, V.T., Thai, N.P., Chea, V., Sam, S., Seng, S., Ding, C. (2016). Introduction of the asian language treebank. In *2016 conference of the oriental chapter of international committee for coordination and standardization of speech databases and assessment techniques (o-cocosda)*(p. 1-6). https://doi.org/10.1109/ICSDA.2016.7918974
Song, X., Li, J., Lei, Q., Zhao, W., Chen, Y., & Mian, A. (2022). Bi-clkt: Bi-graph contrastive learning based knowledge tracing. *Knowledge-Based Systems, 241*, 108274.
Song, X., Li, J., Tang, Y., Zhao, T., Chen, Y., & Guan, Z. (2021). Jkt: A joint graph convolutional network based deep knowledge tracing. *Information Sciences, 580*, 510–523.
Tan, S., Joty, S., Kan, M.-Y., & Socher, R. (2020). It's morphin'time! combating linguistic discrimination with inflectional perturbations. In *Proceedings of the 58th annual meeting of the association for computational linguistics* pp. 2920–2935.
Tang, Y., Tran, C., Li, X., Chen, P.-J., Goyal, N., Chaudhary, V., Gu, V., Fan, A. (2020). Multilingual translation with extensible multilingual pretraining and finetuning.
Tiedemann, J.(2012). Parallel data, tools and interfaces in opus. In *Lrec.*
Tiedemann, J.(2020). The Tatoeba Translation Challenge: Realistic data sets for low resource and multilingual MT. In *Proceedings of the fifth conference on machine translation*(pp. 1174–1182). Association for Computational Linguistics.
Toutanova, K., Klein, D., Manning, C. D., & Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 human language technology conference of the north american chapter of the association for computational linguistics* pp. 252–259.
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., Polosukhin, I. (2017). Attention is all you need Attention is all you need. In *Advances in neural information processing systems* pp. 5998–6008.
Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. (2018). Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 emnlp workshop blackboxnlp: Analyzing and interpreting neural networks for nlp* pp. 353–355.
Wang, J., Xu, C., Guzmán, F., El-Kishky, A., Tang, Y., Rubinstein, B., & Cohn, T. (2021). A targeted attack on neural machine translation using monolingual data poisoning. In *Findings of the association for computational linguistics: Acl-ijcnlp 2021* pp. 1463–1473.
Xu, C., Wang, J., Tang, Y., Guzmán, F., Rubinstein, B. I., & Cohn, T. (2021). A targeted attack on black-box neural machine translation with parallel data poisoning. In *Proceedings of the web conference 2021* pp. 3638–3650.
Xue, G., Zhong, M., Li, J., Chen, J., Zhai, C., & Kong, R. (2022). Dynamic network embedding survey. *Neurocomputing, 472*, 212–223.
Yang, X., Liu, W., Bailey, J., Tao, D., & Liu, W. (2021). Bigram and unigram based text attack via adaptive monotonic heuristic search. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 35, pp. 706–714).
Yang, X., Liu, W., Zhang, S., Liu, W., & Tao, D. (2020). Targeted attention attack on deep learning models in road sign recognition. *IEEE Internet of Things Journal, 86*, 4980–4990.
Yang, Y., Cer, D., Ahmad, A., Guo, M., Law, J., Constant, N., Abrego, G.H., Yuan, S., Tar, C., Sung, Y.H., Strope, B. (2019). Multilingual universal sentence encoder for semantic retrieval. *arXiv preprint*arXiv:1907.04307.
Yin, Z., Wang, F., Liu, W., & Chawla, S. (2018). Sparse feature attacks in adversarial learning. *IEEE Transactions on Knowledge and Data Engineering, 306*, 1164–1177.

Zang, Y., Qi, F., Yang, C., Liu, Z., Zhang, M., Liu, Q., & Sun, M. (2020). Word-level textual adversarial attacking as combinatorial optimization. In *Proceedings of the 58th annual meeting of the association for computational linguistics* pp. 6066–6080.

Zellers, R., Bisk, Y., Schwartz, R., & Choi, Y. (2018). Swag: A large-scale adversarial dataset for grounded commonsense inference. In *Emnlp*.