TRANSLOWDOWN: EFFICIENCY ATTACKS ON NEURAL MACHINE TRANSLATION SYSTEMS

Anonymous authors
Paper under double-blind review

1 GENERAL RESPONSE

1. Configuration of the Maximum Length.

The three victim models in our paper are downloaded from the well-known model repository HuggingFace (https://huggingface.co/), and we adopt the default maximum length configuration from their configuration files. We set the maximum length to 200 in our evaluation, which is the minimum value among the three models (T5 200, FAIR 200, H-NLP 512). To show the effectiveness of our attack under the different maximum length settings, we had performed an experiment, where the victim NMT models set the maximum length to three times of the input length(we observe that the maximum length ratio of the target and the source sentences in training data is more than 2). The experimental results are shown in Table 1. The results in Table 1 show NMT models are vulnerable to efficiency attacks with different maximum length settings (even the computational resource leakage is not as severe as the constant maximum length setting), where C and T represent character-level and token-level attack respectively.

Table 1: I-FLOPs under different maximum length setting

Perturbation	H-NLP		FairSeq		T5	
	C	T	C	T	C	T
1	115.36	162.07	29.34	20.51	79.93	115.46
2	151.96	188.92	51.49	30.55	91.12	116.14
3	162.86	191.84	66.75	36.54	96.96	116.14
4	165.25	192.20	77.02	40.66	100.05	116.14
5	165.65	192.29	82.19	45.48	101.13	116.14

2 RESPONSE TO REVIEWER 1

1. Meaningful of the Proposed Attack

Our work shares the motivation of accuracy-based adversarial attacks, and has unique real-world impact, because of the following reasons:

1. Neural Machine Translation models are more commonly used than traditional machine translation models because NMT models can capture **long dependencies** in sentence. However, the ability to handle long dependencies brings in a new risk, dead loops. There are two mechanisms to avoid dead loops in NMT models, *i.e.,* (i) set a constant maximum length, (ii) set the maximum length according to the input length. However, the effectiveness of these two mechanisms against In this paper, we apply the first mechanism and set the maximum length according to the default configuration files. We evaluate the effectiveness of efficiency adversarial examples under the second maximum length setting and show the results in Table 1.

2. One of the main goals of investigating vulnerabilities is to raise the community concerns, and the fixes for the vulnerabilities are usually straightforward once the vulnerabilities are exposed, *e.g.,* preventing buffer overflow simply requires checking memory boundary when writing unsafe memory. In machine learning community, accuracy-based adversarial attacks have already raised the concern of the committee and the committee is working on defense against accuracy-based attacks. However,

efficiency-based adversarial attacks are currently ignored by the research academia and industry at the current stage. We study the online translation service of HuggingFace (https://huggingface.co/), which is a commercial company that provides the online NLP model API. We randomly select 100 back-end NMT models from HuggingFace's API services [1] and parse each NMT model's configuration file to check how they set the maximum length. Unfortunately, the selected models all set a constant maximum length, and the maximum length is larger than the maximum length in our evaluation (*i.e.,*, the maximum length range from 200 to 1024). In this paper, we first characterize this new vulnerabilities and want to raise the concern of the committee.

2. Configuration of the Maximum Length

See general response.

3 RESPONSE TO REVIEWER 2

1. Apply Output Length as the Evaluation Metric

Table 2: Output Length Increment

NMT	Perturbation Size	Character-Level		Token-Level	
		Baseline	Ours	Baseline	Ours
H-NLP	1	17.33	**337.26**	3.64	**1080.33**
	2	17.33	**764.83**	5.24	**1878.16**
	3	17.33	**960.70**	11.19	**2034.59**
	4	17.33	**1037.64**	15.35	**2066.17**
	5	17.33	**1068.72**	20.20	**2075.20**
FairSeq	1	0.26	**34.48**	0.14	**20.57**
	2	0.26	**83.51**	-0.61	**32.82**
	3	0.26	**138.36**	-2.66	**41.87**
	4	0.26	**181.06**	-6.02	**52.18**
	5	0.26	**211.91**	-9.49	**70.66**
T5	1	4.34	**198.93**	11.20	**316.38**
	2	4.32	**229.85**	11.52	**324.25**
	3	4.32	**246.69**	8.17	**324.25**
	4	4.32	**255.08**	2.00	**324.25**
	5	4.32	**258.70**	-8.25	**324.25**

.

The evaluation results of applying output length as the metric are listed in Table 2. The results are consistent with the metric I-FLOPs. For each generated token, the number of FLOPs in each decoder call is constant, so the results are consistent.

2. Configuration of the Maximum Length

See general response.

3. The effectiveness of attacking online NMT models.

Considering the legal and ethical factors, we did not conduct online experiments. However, the evaluation in this paper is conducted on the NMT models downloaded from HuggingFace (https://huggingface.co/). The victim models are the back-end models that are used for online translation services. Each victim model corresponds to an online translation service on the HuggingFace website (we list the URL of the translation service in Table 2 of the paper). We download the NMT models from the websites to local machines directly, then conduct the experiments. Theoretically, our experiment results indicate that the generated efficiency adversarial examples can also slow down the HuggingFace online translation service. We provide some generated adversarial examples that can be used to test the online URL https://huggingface.co/Helsinki-NLP/opus-mt-en-de.

[1] https://huggingface.co/models?pipeline_tag=translation&sort=downloads

4 RESPONSE TO REVIEWER 3

1. Similarity of the Source Inputs and the Translated Outputs

Table 3: Similarity Between Benign and Adversarial Examples

Subjects	Perturbation	Character-Level		Token-Level	
		Input Similarity	Output Similarity	Input Similarity	Output Similarity
H-NLP	**1**	0.47	0.09	0.63	0.04
	2	0.34	0.04	0.52	0.01
	3	0.30	0.04	0.51	0.01
	4	0.29	0.03	0.51	0.01
	5	0.29	0.03	0.51	0.01
FairSeq	**1**	0.74	0.56	0.82	0.55
	2	0.60	0.41	0.71	0.42
	3	0.52	0.33	0.64	0.36
	4	0.47	0.28	0.58	0.31
	5	0.44	0.26	0.55	0.28
T5	**1**	0.82	0.41	0.89	0.10
	2	0.77	0.34	0.89	0.10
	3	0.75	0.31	0.89	0.10
	4	0.74	0.29	0.89	0.10
	5	0.74	0.28	0.89	0.10

We measure the similarity of benign inputs and adversarial inputs using the BLEU-4 score. The BLEU-4 scores are listed in Table 3. From the results, we observe that the BLEU scores between benign and adversarial inputs are quite high, which indicates the adversarial examples are similar to the benign inputs. However, the BLEU scores between benign and adversarial outputs are very low, the results indicate the efficiency-based adversarial examples will also affect the NMT model accuracy.

2. Overhead of the Attack Algorithms.

Our attack algorithm does not cost many overheads, because it only iterates a limited number of times (the iteration number is equal to the maximum perturbation size). Although we mutate the benign examples to generate some adversarial candidates in each iteration, we batch the generated candidates to select an optimal one with the help of GPU parallelism. The overhead of this process is only a little larger than the overhead of querying the victim NMT models. The overhead results are listed in Table 4, where C and T represent character-level and token-level attack respectively.

Table 4: Overhead of the proposed attacks (s)

Perturbation	H-NLP		FairSeq		T5	
	C	T	C	T	C	T
1	4.01	3.49	4.73	5.64	28.30	33.60
2	8.78	7.70	12.70	11.60	55.40	72.80
3	15.00	12.80	23.10	17.70	82.10	113.00
4	22.30	17.80	33.60	24.00	113.00	151.00
5	29.80	23.00	44.10	30.40	142.00	185.00

3. Insight of Eq 2.

The intuition of Eq 2 comes from two perspectives: *(i)* The goal of the efficiency adversarial examples is to increase the output length to waste the computational consumption. The output length of NMT models is determined by the likelihood of EOS tokens, thus, our first objective is to decrease the likelihood of EOS in order to delay the appearance of EOS. The first objective can be formulated as minimize $\frac{1}{n}\sum_i^n p_i^{eos}$ *(ii)* At the beginning of the optimization, $p_1^{eos}, p_2^{eos}, \cdots, p_{n-1}^{eos}$ are usually small while p_n^{eos} is large. So $o1, o_2, \cdots, o_{n-1}$ keep the same at the beginning of the optimization. However, the process of NMT models generate output tokens is a Markov process *i.e.*,, $p_i = \mathcal{F}_{decoder}(o_{i-1}, h)$. If o_{n-1} keeps the same, modify the inputs only affect h, to accelerate the optimization process, we seek to modify leave the original output $o_1, o_2, \cdots, o_{n-1}$. The second objective can be formulated as minimize $\frac{1}{n}\sum_i^n p_i^{o_i}$ Combining the above two objectives, we have the final objective function in Eq 2.

4. Ensure the Adversarial Perturbation less than the Maximum Perturbation

At each iteration, we mutate the input with one perturbation size, thus, the iteration number will limit the adversarial perturbation size. If the maximum perturbation is set as 5, we just iterate 5 times to generate the adversarial perturbations.

5. UNK Related Issues

It is fine for our algorithm if the corrupted result is tokenized into multiple tokens or the UNK token. Because for the next iteration, our algorithm will apply Equation 2 to find the new important tokens in the mutated sentences. If the UNK token are the most important tokens in the new sentences, our algorithm will mutate the UNK token and select a optimal candidates.

6. Defense of the Proposed Efficiency Attacks

The generated efficiency adversarial examples can be used for adversarial training to increase the NMT models efficiency robustness. As shown in Table 3, the efficiency adversarial examples also decrease the accuracy of the NMT models. Thus, If we feed the adversarial inputs and the true target outputs to the NMT models, NMT models robustness can be improved.

5 RESPONSE TO REVIEWER 4

1. Configuration of the Maximum Length

See general response.

2. Evaluation on Other Model Architectures

We evaluate our proposed attacks on two more model architectures (LSTM and GRU encoder and decoder). The results are listed in Table 5, the results show the proposed attacks also slowdown the LSTM and GRU based NMT models.

Table 5: Results on other model architectures

Perturbation	LSTM		GRU	
	C	T	C	T
1	56.43	144.42	25.43	220.32
2	57.92	155.43	39.54	250.32
3	143.20	166.43	44.54	288.32
4	166.42	177.32	88.43	320.32
5	220.32	192.29	90.54	343.69