

# On Evaluation of Adversarial Perturbations for Sequence-to-Sequence Models

Paul Michel<sup>1</sup>, Xian Li<sup>2</sup>, Graham Neubig<sup>1</sup>, Juan Miguel Pino<sup>2</sup>

<sup>1</sup>Language Technologies Institute, Carnegie Mellon University

<sup>2</sup>Facebook AI

{pmichell, gneubig}@cs.cmu.edu,

{xianl, juancarabina}@fb.com

## Abstract

Adversarial examples — perturbations to the input of a model that elicit large changes in the output — have been shown to be an effective way of assessing the robustness of *sequence-to-sequence* (seq2seq) models. However, these perturbations only indicate weaknesses in the model if they do not change the input so significantly that it legitimately results in changes in the expected output. This fact has largely been ignored in the evaluations of the growing body of related literature. Using the example of untargeted attacks on *machine translation* (MT), we propose a new evaluation framework for adversarial attacks on seq2seq models that takes the semantic equivalence of the pre- and post-perturbation input into account. Using this framework, we demonstrate that existing methods may not preserve meaning in general, breaking the aforementioned assumption that source side perturbations should not result in changes in the expected output. We further use this framework to demonstrate that adding additional constraints on attacks allows for adversarial perturbations that are more meaning-preserving, but nonetheless largely change the output sequence. Finally, we show that performing untargeted adversarial training with meaning-preserving attacks is beneficial to the model in terms of adversarial robustness, without hurting test performance.<sup>1</sup>

## 1 Introduction

Attacking a machine learning model with adversarial perturbations is the process of making changes to its input to maximize an adversarial goal, such as mis-classification (Szegedy et al., 2013) or mis-translation (Zhao et al., 2018). These attacks provide insight into the vulnerabilities of machine learning models and their brittleness to

<sup>1</sup>A toolkit implementing our evaluation framework is released at <https://github.com/pmichel31415/teapot-nlp>.

samples outside the training distribution. Lack of robustness to these attacks poses security concerns to safety-critical applications, *e.g.* self-driving cars (Bojarski et al., 2016).

Adversarial attacks were first defined and investigated for computer vision systems (Szegedy et al. (2013); Goodfellow et al. (2014); Moosavi-Dezfooli et al. (2016) *inter alia*), where the input space is continuous, making minuscule perturbations largely imperceptible to the human eye. In discrete spaces such as natural language sentences, the situation is more problematic; even a flip of a single word or character is generally perceptible by a human reader. Thus, most of the mathematical framework in previous work is not directly applicable to discrete text data. Moreover, there is no canonical distance metric for textual data like the  $\ell_p$  norm in real-valued vector spaces such as images, and evaluating the level of semantic similarity between two sentences is a field of research of its own (Cer et al., 2017). This elicits a natural question: *what does the term “adversarial perturbation” mean in the context of natural language processing (NLP)?*

We propose a simple but natural criterion for adversarial examples in NLP, particularly untargeted<sup>2</sup> attacks on seq2seq models: *adversarial examples should be meaning-preserving on the source side, but meaning-destroying on the target side.* The focus on explicitly evaluating meaning preservation is in contrast to previous work on adversarial examples for seq2seq models (Belinkov and Bisk, 2018; Zhao et al., 2018; Cheng et al., 2018; Ebrahimi et al., 2018a). Nonetheless, this feature is extremely important; given two sentences with equivalent meaning, we would expect a good model to produce two outputs with

<sup>2</sup>Here we use the term untargeted in the same sense as (Ebrahimi et al., 2018a): an attack whose goal is simply to decrease performance with respect to a reference translation.

equivalent meaning. In other words, any meaning-preserving perturbation that results in the model output changing drastically highlights a fault of the model.

A first technical contribution of this paper is to lay out a method for formalizing this concept of meaning-preserving perturbations (§2). This makes it possible to evaluate the effectiveness of adversarial attacks or defenses either using gold-standard human evaluation, or approximations that can be calculated without human intervention. We further propose a simple method of imbuing gradient-based word substitution attacks (§3.1) with simple constraints aimed at increasing the chance that the meaning is preserved (§3.2).

Our experiments are designed to answer several questions about meaning preservation in seq2seq models. First, we evaluate our proposed “source-meaning-preserving, target-meaning-destroying” criterion for adversarial examples using both manual and automatic evaluation (§4.2) and find that a less widely used evaluation metric (chrF) provides significantly better correlation with human judgments than the more widely used BLEU and METEOR metrics. We proceed to perform an evaluation of adversarial example generation techniques, finding that chrF does help to distinguish between perturbations that are more meaning-preserving across a variety of languages and models (§4.3). Finally, we apply existing methods for adversarial training to the adversarial examples with these constraints and show that making adversarial inputs more semantically similar to the source is beneficial for robustness to adversarial attacks and does not decrease test performance on the original data distribution (§5).

## 2 A Framework for Evaluating Adversarial Attacks

In this section, we present a simple procedure for evaluating adversarial attacks on seq2seq models. We will use the following notation:  $x$  and  $y$  refer to the source and target sentence respectively. We denote  $x$ ’s translation by model  $M$  as  $y_M$ . Finally,  $\hat{x}$  and  $\hat{y}_M$  represent an adversarially perturbed version of  $x$  and its translation by  $M$ , respectively. The nature of  $M$  and the procedure for obtaining  $\hat{x}$  from  $x$  are irrelevant to the discussion below.

### 2.1 The Adversarial Trade-off

The goal of adversarial perturbations is to produce failure cases for the model  $M$ . Hence, the evaluation must include some measure of the *target similarity* between  $y$  and  $y_M$ , which we will denote  $s_{\text{tgt}}(y, \hat{y}_M)$ . However, if no distinction is being made between perturbations that preserve the meaning and those that don’t, a sentence like “he’s very *friendly*” is considered a valid adversarial perturbation of “he’s very *adversarial*”, even though its meaning is the opposite. Hence, it is crucial, when evaluating adversarial attacks on MT models, that the discrepancy between the original and adversarial input sentence be quantified in a way that is sensitive to meaning. Let us denote such a *source similarity* score  $s_{\text{src}}(x, \hat{x})$ .

Based on these functions, we define the *target relative score decrease* as:

$$d_{\text{tgt}}(y, y_M, \hat{y}_M) = \begin{cases} 0 & \text{if } s_{\text{tgt}}(y, \hat{y}_M) \geq s_{\text{tgt}}(y, y_M) \\ \frac{s_{\text{tgt}}(y, y_M) - s_{\text{tgt}}(y, \hat{y}_M)}{s_{\text{tgt}}(y, y_M)} & \text{otherwise} \end{cases} \quad (1)$$

The choice to report the *relative* decrease in  $s_{\text{tgt}}$  makes scores comparable across different models or languages<sup>3</sup>. For instance, for languages that are comparatively easy to translate (e.g. French-English),  $s_{\text{tgt}}$  will be higher in general, and so will the gap between  $s_{\text{tgt}}(y, y_M)$  and  $s_{\text{tgt}}(y, \hat{y}_M)$ . However this does not necessarily mean that attacks on this language pair are more effective than attacks on a “difficult” language pair (e.g. Czech-English) where  $s_{\text{tgt}}$  is usually smaller.

We recommend that both  $s_{\text{src}}$  and  $d_{\text{tgt}}$  be reported when presenting adversarial attack results. However, in some cases where a single number is needed, we suggest reporting the attack’s *success*  $\mathcal{S} := s_{\text{src}} + d_{\text{tgt}}$ . The interpretation is simple:  $\mathcal{S} > 1 \Leftrightarrow d_{\text{tgt}} > 1 - s_{\text{src}}$ , which means that the attack has destroyed the target meaning ( $d_{\text{tgt}}$ ) more than it has destroyed the source meaning ( $1 - s_{\text{src}}$ ).

Importantly, this framework can be extended beyond strictly meaning-preserving attacks. For example, for targeted keyword introduction attacks (Cheng et al., 2018; Ebrahimi et al., 2018a), the same evaluation framework can be used if  $s_{\text{tgt}}$  (resp.  $s_{\text{src}}$ ) is modified to account for the presence (resp. absence) of the keyword (or its translation in the source). Similarly this can be extended to other

<sup>3</sup>Note that we do not allow negative  $d_{\text{tgt}}$  to keep all scores between 0 and 1.

tasks by adapting  $s_{tgt}$  (e.g. for classification one would use the zero-one loss, and adapt the success threshold).

## 2.2 Similarity Metrics

Throughout §2.1, we have not given an exact description of the semantic similarity scores  $s_{src}$  and  $s_{tgt}$ . Indeed, automatically evaluating the semantic similarity between two sentences is an open area of research and it makes sense to decouple the definition of adversarial examples from the specific method used to measure this similarity. In this section, we will discuss manual and automatic metrics that may be used to calculate it.

### 2.2.1 Human Judgment

Judgment by speakers of the language of interest is the *de facto* gold standard metric for semantic similarity. Specific criteria such as adequacy/fluency (Ma and Cieri, 2006), acceptability (Goto et al., 2013), and 6-level semantic similarity (Cer et al., 2017) have been used in evaluations of MT and sentence embedding methods. In the context of adversarial attacks, we propose the following 6-level evaluation scheme, which is motivated by previous measures, but designed to be (1) symmetric, like Cer et al. (2017), (2) and largely considers meaning preservation but at the very low and high levels considers fluency of the output<sup>4</sup>, like Goto et al. (2013):

How would you rate the similarity between the meaning of these two sentences?

0. The meaning is completely different or one of the sentences is meaningless
1. The topic is the same but the meaning is different
2. Some key information is different
3. The key information is the same but the details differ
4. Meaning is essentially equal but some expressions are unnatural
5. Meaning is essentially equal and the two sentences are well-formed English<sup>a</sup>

<sup>a</sup>Or the language of interest.

<sup>4</sup>This is important to rule out nonsensical sentences and distinguish between clean and “noisy” paraphrases (e.g. typos, non-native speech...). We did not give annotators additional instruction specific to typos.

### 2.2.2 Automatic Metrics

Unfortunately, human evaluation is expensive, slow and sometimes difficult to obtain, for example in the case of low-resource languages. This makes automatic metrics that do not require human intervention appealing for experimental research. This section describes 3 evaluation metrics commonly used as alternatives to human evaluation, in particular to evaluate translation models.<sup>5</sup>

**BLEU:** (Papineni et al., 2002) is an automatic metric based on n-gram precision coupled with a penalty for shorter sentences. It relies on exact word-level matches and therefore cannot detect synonyms or morphological variations.

**METEOR:** (Denkowski and Lavie, 2014) first estimates alignment between the two sentences and then computes unigram F-score (biased towards recall) weighted by a penalty for longer sentences. Importantly, METEOR uses stemming, synonymy and paraphrasing information to perform alignments. On the downside, it requires language specific resources.

**chrF:** (Popović, 2015) is based on the character n-gram F-score. In particular we will use the chrF2 score (based on the F2-score — recall is given more importance), following the recommendations from Popović (2016). By operating on a sub-word level, it can reflect the semantic similarity between different morphological inflections of one word (for instance), without requiring language-specific knowledge which makes it a good one-size-fits-all alternative.

Because multiple possible alternatives exist, it is important to know which is the best stand-in for human evaluation. To elucidate this, we will compare these metrics to human judgment in terms of Pearson correlation coefficient on outputs resulting from a variety of attacks in §4.2.

## 3 Gradient-Based Adversarial Attacks

In this section, we overview the adversarial attacks we will be considering in the rest of this paper.

### 3.1 Attack Paradigm

We perform gradient-based attacks that replace one word in the sentence so as to maximize an adversarial loss function  $\mathcal{L}_{adv}$ , similar to the substitution attacks proposed in (Ebrahimi et al., 2018b).

<sup>5</sup>Note that other metrics of similarity are certainly applicable within the overall framework of §2.2.1, but we limit our examination in this paper to the three noted here.

Original	<b>Pourquoi</b> faire cela ?
English gloss	<b>Why</b> do this?
Unconstrained	<b>construisant</b> (English: building) faire cela ?
kNN	<b>interrogez</b> (English: interrogate) faire cela ?
CharSwap	<b>Puorquoi</b> (typo) faire cela ?
Original	Si seulement je pouvais me muscler <b>aussi</b> rapidement.
English gloss	If only I could build my muscle <b>this</b> fast.
Unconstrained	Si seulement je pouvais me muscler <b>etc</b> rapidement.
kNN	Si seulement je pouvais me muscler <b>plsu</b> (typo for “more”) rapidement.
CharSwap	Si seulement je pouvais me muscler <b>asusi</b> (typo) rapidement.

Table 1: Examples of different adversarial inputs. The substituted word is highlighted.

### 3.1.1 General Approach

Precisely, for a word-based translation model  $M^6$ , and given an input sentence  $w_1, \dots, w_n$ , we find the position  $i^*$  and word  $w^*$  satisfying the following optimization problem:

$$\arg \max_{1 \leq i \leq n, \hat{w} \in \mathcal{V}} \mathcal{L}_{\text{adv}}(w_0, \dots, w_{i-1}, \hat{w}, w_{i+1}, \dots, w_n) \quad (2)$$

where  $\mathcal{L}_{\text{adv}}$  is a differentiable function which represents our adversarial objective. Using the first order approximation of  $\mathcal{L}_{\text{adv}}$  around the original word vectors  $\mathbf{w}_1, \dots, \mathbf{w}_n^7$ , this can be derived to be equivalent to optimizing

$$\arg \max_{1 \leq i \leq n, \hat{w} \in \mathcal{V}} [\hat{\mathbf{w}} - \mathbf{w}_i]^\top \nabla_{\mathbf{w}_i} \mathcal{L}_{\text{adv}} \quad (3)$$

The above optimization problem can be solved by brute-force in  $\mathcal{O}(n|\mathcal{V}|)$  space complexity, whereas the time complexity is bottlenecked by a  $|\mathcal{V}| \times d$  times  $n \times d$  matrix multiplication, which is not more computationally expensive than computing logits during the forward pass of the model. Overall, this naive approach is sufficiently fast to be conducive to adversarial training. We also found that the attacks benefited from normalizing the gradient by taking its sign.

Extending this approach to finding the optimal perturbations for more than 1 substitution would require exhaustively searching over all possible combinations. However, previous work (Ebrahimi

<sup>6</sup>Note that this formulation is also valid for character-based models (see Ebrahimi et al. (2018a)) and subword-based models. For subword-based models, additional difficulty would be introduced due to changes to the input resulting in different subword segmentations. This poses an interesting challenge that is beyond the scope of the current work.

<sup>7</sup>More generally we will use the bold  $\mathbf{w}$  when talking about the embedding vector of word  $w$

et al., 2018a) suggests that greedy search is a good enough approximation.

### 3.1.2 The Adversarial Loss $\mathcal{L}_{\text{adv}}$

We want to find an adversarial input  $\hat{x}$  such that, assuming that the model has produced the correct output  $y_1, \dots, y_{t-1}$  up to step  $t - 1$  during decoding, the probability that the model makes an error at the next step  $t$  is maximized.

In the log-semiring, this translates into the following loss function:

$$\mathcal{L}_{\text{adv}}(\hat{x}, y) = \sum_{t=1}^{|y|} \log(1 - p(y_t | \hat{x}, y_1, \dots, y_{t-1})) \quad (4)$$

## 3.2 Enforcing Semantically Similar Adversarial Inputs

In contrast to previous methods, which don’t consider meaning preservation, we propose simple modifications of the approach presented in §3.1 to create adversarial perturbations at the word level that are more likely to preserve meaning. The basic idea is to restrict the possible word substitutions to similar words. We compare two sets of constraints:

**kNN:** This constraint enforces that the word be replaced only with one of its 10 nearest neighbors in the source embedding space. This has two effects: first, the replacement will be likely semantically related to the original word (if words close in the embedding space are indeed semantically related, as hinted by Table 1). Second, it ensures that the replacement’s word vector is close enough to the original word vector that the first order assumption is more likely to be satisfied.

**CharSwap:** This constraint requires that the substituted words must be obtained by swapping

characters. Word internal character swaps have been shown to not affect human readers greatly (McCusker et al., 1981), hence making them likely to be meaning-preserving. Moreover we add the additional constraint that the substitution must not be in the vocabulary, which will likely be particularly meaning-destroying on the target side for the word-based models we test here. In such cases where word-internal character swaps are not possible or can't produce **out-of-vocabulary (OOV)** words, we resort to the naive strategy of repeating the last character of the word. The exact procedure used to produce this kind of perturbations is described in Appendix A.1. Note that for a word-based model, every **OOV** will look the same (a special `<unk>` token), however the choice of **OOV** will still have an influence on the output of the model because we use unk-replacement.

In contrast, we refer the base attack without constraints as **Unconstrained** hereforth. Table 1 gives qualitative examples of the kind of perturbations generated under the different constraints.

For subword-based models, we apply the same procedures at the subword-level on the original segmentation. We then de-segment and re-segment the resulting sentence (because changes at the subword or character levels are likely to change the segmentation of the resulting sentence).

## 4 Experiments

Our experiments serve two purposes. First, we examine our proposed framework of evaluating adversarial attacks (§2), and also elucidate which automatic metrics correlate better with human judgment for the purpose of evaluating adversarial attacks (§4.2). Second, we use this evaluation framework to compare various adversarial attacks and demonstrate that adversarial attacks that are explicitly constrained to preserve meaning receive better assessment scores (§4.3).

### 4.1 Experimental setting

**Data:** Following previous work on adversarial examples for `seq2seq` models (Blinkov and Bisk, 2018; Ebrahimi et al., 2018a), we perform all experiments on the IWSLT2016 dataset (Cettolo et al., 2016) in the {French, German, Czech}→English directions (fr-en, de-en and cs-en). We compile all previous IWSLT test sets before 2015 as

validation data, and keep the 2015 and 2016 test sets as test data. The data is tokenized with the Moses tokenizer (Koehn et al., 2007). The exact data statistics can be found in Appendix A.2.

**MT Models:** We perform experiments with two common **neural machine translation (NMT)** models. The first is an LSTM based encoder-decoder architecture with attention (Luong et al., 2015). It uses 2-layer encoders and decoders, and dot-product attention. We set the word embedding dimension to 300 and all others to 500. The second model is a self-attentional Transformer (Vaswani et al., 2017), with 6 1024-dimensional encoder and decoder layers and 512 dimensional word embeddings. Both the models are trained with Adam (Kingma and Ba, 2014), dropout (Srivastava et al., 2014) of probability 0.3 and label smoothing (Szegedy et al., 2016) with value 0.1. We experiment with both word based models (vocabulary size fixed at 40k) and subword based models (BPE (Sennrich et al., 2016) with 30k operations). For word-based models, we perform `<unk>` replacement, replacing `<unk>` tokens in the translated sentences with the source words with the highest attention value during inference. The full experimental setup and source code are available at [https://github.com/pmichel31415/translate/tree/paul/pytorch\\_translate/research/adversarial/experiments](https://github.com/pmichel31415/translate/tree/paul/pytorch_translate/research/adversarial/experiments).

**Automatic Metric Implementations:** To evaluate both sentence and corpus level BLEU score, we first de-tokenize the output and use `sacreBLEU`<sup>8</sup> (Post, 2018) with its internal `intl` tokenization, to keep BLEU scores agnostic to tokenization. We compute METEOR using the official implementation<sup>9</sup>. ChrF is reported with the `sacreBLEU` implementation on detokenized text with default parameters. A toolkit implementing the evaluation framework described in §2.1 for these metrics is released at <https://github.com/pmichel31415/teapot-nlp>.

### 4.2 Correlation of Automatic Metrics with Human Judgment

We first examine which of the automatic metrics listed in §2.2 correlates most with human judgment for our adversarial attacks. For this experiment, we restrict the scope to the case of the

<sup>8</sup><https://github.com/mjpost/sacreBLEU>

<sup>9</sup><http://www.cs.cmu.edu/~alavie/METEOR/>

		LSTM			Transformer		
Language pair		cs-en	de-en	fr-en	cs-en	de-en	fr-en
Word-based		Target RDChrF			Target RDChrF		
	Original chrF	45.68	49.43	57.49	47.66	51.08	58.04
	Unconstrained	25.38	25.54	25.59	25.24	25.00	24.68
	CharSwap	24.11	24.94	23.60	21.59	23.23	21.75
	kNN	15.00	15.59	15.22	20.74	19.97	18.59
		Source chrF			Source chrF		
	Unconstrained	70.14	72.39	74.29	69.03	71.93	73.23
	CharSwap	82.65	84.40	86.62	84.13	85.97	87.02
	kNN	78.08	78.11	77.62	74.94	77.92	77.88
	Subword-based		Target RDChrF			Target RDChrF	
Original chrF		48.30	52.42	59.08	49.70	54.01	59.65
Unconstrained		25.79	26.03	26.96	23.97	25.07	25.28
CharSwap		18.65	19.15	19.75	16.98	18.38	17.85
kNN		15.00	16.26	17.12	19.02	18.58	18.63
		Source chrF			Source chrF		
Unconstrained		69.32	72.12	73.57	68.66	71.51	72.65
CharSwap		85.84	87.46	87.98	85.79	87.07	87.99
kNN		76.17	77.74	78.03	73.05	75.91	76.54

Table 2: Target RDchrF and source chrF scores for all the attacks on all our models (word- and subword-based LSTM and Transformer).

LSTM model on `fr-en`. For the French side, we randomly select 900 sentence pairs  $(x, \hat{x})$  from the validation set, 300 for each of the Unconstrained, kNN and CharSwap constraints. To vary the level of perturbation, the 300 pairs contain an equal amount of perturbed input obtained by substituting 1, 2 and 3 words. On the English side, we select 900 pairs of reference translations and translations of adversarial input  $(y, \hat{y}_M)$  with the same distribution of attacks as the source side, as well as 300  $(y, y_M)$  pairs (to include translations from original inputs). This amounts to 1,200 sentence pairs in the target side.

These sentences are sent to English and French speaking annotators to be rated according to the guidelines described in §2.2.1. Each sample (a pair of sentences) is rated by two independent evaluators. If the two ratings differ, the sample is sent to a third rater (an auditor and subject matter expert) who makes the final decision.

Finally, we compare the human results to each automatic metric with Pearson’s correlation coefficient. The correlations are reported in Table 3. As evidenced by the results, chrF exhibits higher correlation with human judgment, followed by METEOR and BLEU. This is true both on the source side  $(x$  vs  $\hat{x})$  and in the target side  $(y$  vs  $\hat{y}_M)$ . We

Language	BLEU	METEOR	chrF
French	0.415	0.440	<b>0.586*</b>
English	0.357	0.478*	<b>0.497</b>

Table 3: Correlation of automatic metrics to human judgment of adversarial source and target sentences. “\*” indicates that the correlation is significantly better than the next-best one.

evaluate the statistical significance of this result using a paired bootstrap test for  $p < 0.01$ . Notably we find that chrF is significantly better than METEOR in French but not in English. This is not too unexpected because METEOR has access to more language-dependent resources in English (specifically synonym information) and thereby can make more informed matches of these synonymous words and phrases. Moreover the French source side contains more “character-level” errors (from CharSwap attacks) which are not picked-up well by word-based metrics like BLEU and METEOR. For a breakdown of the correlation coefficients according to number of perturbation and type of constraints, we refer to Appendix A.3.

Thus, in the following, we report attack results both in terms of chrF in the source ( $s_{src}$ ) and **relative decrease in chrF (RDchrF)** in the target ( $d_{tgt}$ ).

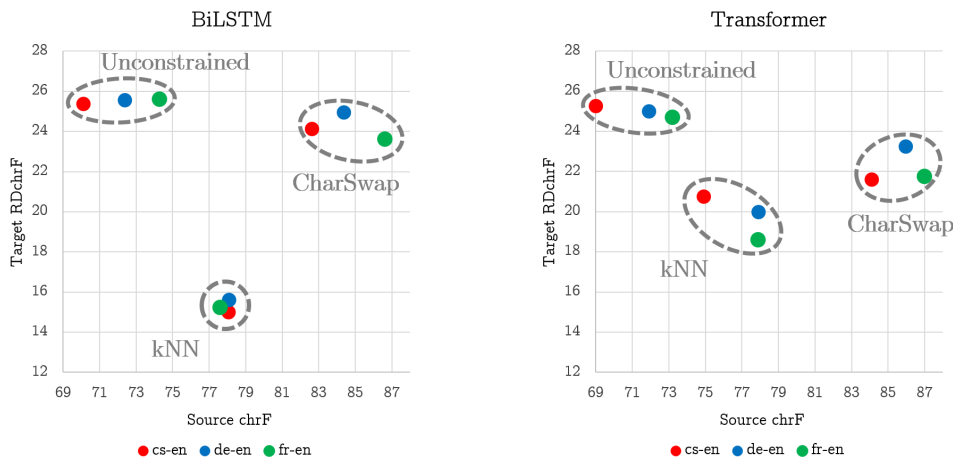


Figure 1: Graphical representation of the results in Table 2 for word-based models. High source chrF and target RDchrF (upper-right corner) indicates a good attack.

### 4.3 Attack Results

We can now compare attacks under the three constraints Unconstrained, kNN and CharSwap and draw conclusions on their capacity to preserve meaning in the source and destroy it in the target. Attacks are conducted on the validation set using the approach described in §3.1 with 3 substitutions (this means that each adversarial input is at edit distance at most 3 from the original input). Results (on a scale of 0 to 100 for readability) are reported in Table 2 for both word- and subword-based LSTM and Transformer models. To give a better idea of how the different variables (language pair, model, attack) affect performance, we give a graphical representation of these same results in Figure 1 for the word-based models. The rest of this section discusses the implication of these results.

**Source chrF Highlights the Effect of Adding Constraints:** Comparing the kNN and CharSwap rows to Unconstrained in the “source” sections of Table 2 clearly shows that constrained attacks have a positive effect on meaning preservation. Beyond validating our assumptions from §3.2, this shows that source chrF is useful to carry out the comparison in the first place<sup>10</sup>. To give a point of reference, results from the manual evaluation carried out in §4.2 show that that 90% of the French sentence pairs to which humans gave a score of 4 or 5 in semantic similarity have a chrF > 78.

<sup>10</sup>It can be argued that using chrF gives an advantage to CharSwap over kNN for source preservation (as opposed to METEOR for example). We find that this is the case for Czech and German (source METEOR is higher for kNN) but not French. Moreover we find (see A.3) that chrF correlates better with human judgement even for kNN.

Successful attack (source chrF = 80.89, target RDchrF = 84.06)	
Original	Ils le réinvestissent directement en engageant plus de procès.
Adv. src	<b>I</b> ls le réinvestissent <b>dierctement</b> en <b>engagaent</b> plus de procès.
Ref.	They plow it right back into filing more troll lawsuits.
Base output	They direct it directly by engaging more cases.
Adv. output	.. de plus.
Unsuccessful attack (source chrF = 54.46, target RDchrF = 0.00)	
Original	C’était en Juillet 1969.
Adv. src	C’ <b>étiat</b> en <b>Jiullet</b> 1969.
Ref.	This is from July, 1969.
Base output	This was in July 1969.
Adv. output	This is. in 1969.

Table 4: Example of CharSwap attacks on the fr-en LSTM. The first example is a successful attack (high source chrF and target RDchrF) whereas the second is not.

**Different Architectures are not Equal in the Face of Adversity:** Inspection of the target-side results yields several interesting observations. First, the high RDchrF of CharSwap for word-based model is yet another indication of their known shortcomings when presented with words out of their training vocabulary, even with <unk>-replacement. Second, and perhaps more interestingly, Transformer models appear to be less robust to small embedding perturbations (kNN attacks) compared to LSTMs. Although the exploration of the exact reasons for this phenomenon is beyond the scope of this work, this is a good example that RDchrF can shed light on the different behavior of different architectures when confronted with adversarial input. Overall, we find that the Char-

Swap constraint is the only one that consistently produces attacks with  $> 1$  average success (as defined in Section 2.1) according to Table 2. Table 4 contains two qualitative examples of this attack on the LSTM model in fr-en.

## 5 Adversarial Training with Meaning-Preserving Attacks

### 5.1 Adversarial Training

Adversarial training (Goodfellow et al., 2014) augments the training data with adversarial examples. Formally, in place of the **negative log likelihood (NLL)** objective on a sample  $x, y$ ,  $\mathcal{L}(x, y) = NLL(x, y)$ , the loss function is replaced with an interpolation of the **NLL** of the original sample  $x, y$  and an adversarial sample  $\hat{x}, y$ :

$$\mathcal{L}'(x, y) = (1 - \alpha)NLL(x, y) + \alpha NLL(\hat{x}, y) \quad (5)$$

Ebrahimi et al. (2018a) suggest that while adversarial training improves robustness to adversarial attacks, it can be detrimental to test performance on non-adversarial input. We investigate whether this is still the case when adversarial attacks are largely meaning-preserving.

In our experiments, we generate  $\hat{x}$  by applying 3 perturbations on the fly at each training step. To maintain training speed we do not solve Equation (2) iteratively but in one shot by replacing the argmax by top-3. Although this is less exact than iterating, this makes adversarial training time less than  $2\times$  slower than normal training. We perform adversarial training with perturbations without constraints (Unconstrained-adv) and with the CharSwap constraint (CharSwap-adv). All experiments are conducted with the word-based LSTM model.

### 5.2 Results

Test performance on non-adversarial input is reported in Table 5. In keeping with the rest of the paper, we primarily report chrF results, but also show the standard BLEU as well.

We observe that when  $\alpha = 1.0$ , *i.e.* the model only sees the perturbed input during training<sup>11</sup>, the Unconstrained-adv model suffers a drop in test performance, whereas CharSwap-adv’s performance is on par with the original. This is likely

<sup>11</sup>This setting is reminiscent of word dropout (Iyyer et al., 2015).

Language pair	cs-en	de-en	fr-en
Base	44.21	49.30	55.67
	(22.89)	(28.61)	(35.28)
$\alpha = 1.0$			
Unconstrained-adv	41.38	46.15	53.39
	(21.51)	(27.06)	(33.96)
CharSwap-adv	43.74	48.85	55.60
	(23.00)	(28.45)	(35.33)
$\alpha = 0.5$			
Unconstrained-adv	43.68	48.60	55.55
	(22.93)	(28.30)	(35.25)
CharSwap-adv	44.57	49.14	55.88
	(23.66)	(28.66)	(35.63)

Table 5: chrF (BLEU) scores on the original test set before/after adversarial training of the word-based LSTM model.

Language pair	cs-en	de-en	fr-en
Base	24.11	24.94	23.60
	$\alpha = 1.0$		
Unconstrained-adv	25.99	26.24	25.67
CharSwap-adv	16.46	17.19	15.72
$\alpha = 0.5$			
Unconstrained-adv	26.52	27.26	24.92
CharSwap-adv	20.41	20.24	16.08

Table 6: Robustness to CharSwap attacks on the validation set with/without adversarial training (RDchrF). Lower is better.

attributable to the spurious training samples  $(\hat{x}, y)$  where  $y$  is not an acceptable translation of  $\hat{x}$  introduced by the lack of constraint. This effect disappears when  $\alpha = 0.5$  because the model sees the original samples as well.

Not unexpectedly, Table 6 indicates that CharSwap-adv is more robust to CharSwap constrained attacks for both values of  $\alpha$ , with 1.0 giving the best results. On the other hand, Unconstrained-adv is similarly or more vulnerable to these attacks than the baseline. Hence, we can safely conclude that adversarial training with CharSwap attacks improves robustness while not impacting test performance as much as unconstrained attacks.

## 6 Related work

Following seminal work on adversarial attacks by Szegedy et al. (2013), Goodfellow et al. (2014) introduced gradient-based attacks and adversarial training. Since then, a variety of attack (Moosavi-



Dezfooli et al., 2016) and defense (Cissé et al., 2017; Kolter and Wong, 2017) mechanisms have been proposed. Adversarial examples for NLP specifically have seen attacks on sentiment (Papernot et al., 2016; Samanta and Mehta, 2017; Ebrahimi et al., 2018b), malware (Grosse et al., 2016), gender (Reddy and Knight, 2016) or toxicity (Hosseini et al., 2017) classification to cite a few.

In MT, methods have been proposed to attack word-based (Zhao et al., 2018; Cheng et al., 2018) and character-based (Belinkov and Bisk, 2018; Ebrahimi et al., 2018a) models. However these works side-step the question of meaning preservation in the source: they mostly focus on target side evaluation. Finally there is work centered around meaning-preserving adversarial attacks for NLP via paraphrase generation (Iyyer et al., 2018) or rule-based approaches (Jia and Liang, 2017; Ribeiro et al., 2018; Naik et al., 2018; Alzantot et al., 2018). However the proposed attacks are highly engineered and focused on English.

## 7 Conclusion

This paper highlights the importance of performing *meaning-preserving* adversarial perturbations for NLP models (with a focus on seq2seq). We proposed a general evaluation framework for adversarial perturbations and compared various automatic metrics as proxies for human judgment to instantiate this framework. We then confirmed that, in the context of MT, “naive” attacks do not preserve meaning in general, and proposed alternatives to remedy this issue. Finally, we have shown the utility of adversarial training in this paradigm. We hope that this helps future work in this area of research to evaluate meaning conservation more consistently.

## Acknowledgments

The authors would like to extend their thanks to members of the LATTE team at Facebook and Neulab at Carnegie Mellon University for valuable discussions, as well as the anonymous reviewers for their insightful feedback. This research was partially funded by Facebook.

## References

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang.

2018. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896.

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

Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Praseoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. 2016. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14.

Mauro Cettolo, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, Roldano Cattoni, and Marcello Federico. 2016. The iwslt 2016 evaluation campaign.

Minhao Cheng, Jinfeng Yi, Huan Zhang, Pin-Yu Chen, and Cho-Jui Hsieh. 2018. Seq2sick: Evaluating the robustness of sequence-to-sequence models with adversarial examples. *arXiv preprint arXiv:1803.01128*.

Moustapha Cissé, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. 2017. Parseval networks: Improving robustness to adversarial examples. In *International Conference on Machine Learning*.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*.

Javid Ebrahimi, Daniel Lowd, and Dejing Dou. 2018a. On adversarial examples for character-level neural machine translation. In *COLING*.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018b. Hotflip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Isao Goto, Ka-Po Chow, Bin Lu, Eiichiro Sumita, and Benjamin Ka-Yin T’sou. 2013. Overview of the patent machine translation task at the ntcir-10 workshop. In *NII Test Collection for IR Systems*.

- Kathrin Grosse, Nicolas Papernot, Praveen Manoharan, Michael Backes, and Patrick McDaniel. 2016. Adversarial perturbations against deep neural networks for malware classification. *arXiv preprint arXiv:1606.04435*.
- Hossein Hosseini, Sreeram Kannan, Baosen Zhang, and Radha Poovendran. 2017. Deceiving google’s perspective api built for detecting toxic comments. *arXiv preprint arXiv:1702.08138*.
- Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. 2015. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1681–1691.
- Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1875–1885.
- Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031.
- Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, ACL ’07*, pages 177–180.
- J. Zico Kolter and Eric Wong. 2017. Provable defenses against adversarial examples via the convex outer adversarial polytope. *arXiv preprint arXiv:1711.00851*.
- Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.
- Xiaoyi Ma and Christopher Cieri. 2006. Corpus support for machine translation at ldc. In *Language Resources and Evaluation Conference*.
- Leo X McCusker, Philip B Gough, and Randolph G Bias. 1981. Word recognition inside out and outside in. *Journal of Experimental Psychology: Human Perception and Performance*, 7(3):538.
- Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. 2016. Deepfool: A simple and accurate method to fool deep neural networks. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2574–2582.
- Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353.
- Nicolas Papernot, Patrick McDaniel, Ananthram Swami, and Richard Harang. 2016. Crafting adversarial input sequences for recurrent neural networks. In *Military Communications Conference, MILCOM 2016-2016 IEEE*, pages 49–54. IEEE.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.
- Maja Popović. 2015. chrF: character n-gram f-score for automatic mt evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395.
- Maja Popović. 2016. chrF deconstructed: beta parameters and n-gram weights. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 499–504.
- Matt Post. 2018. A call for clarity in reporting bleu scores. *arXiv preprint arXiv:1804.08771*.
- Sravana Reddy and Kevin Knight. 2016. Obfuscating gender in social media writing. In *Proceedings of the First Workshop on NLP and Computational Social Science*. Association for Computational Linguistics.
- Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically equivalent adversarial rules for debugging nlp models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 856–865.
- Suranjana Samanta and Sameep Mehta. 2017. Towards crafting text adversarial samples. *arXiv preprint arXiv:1707.02812*.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725.

- Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research (JMLR)*, 15(1):1929–1958.
- Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826.
- Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.
- 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 Annual Conference on Neural Information Processing Systems (NIPS)*, pages 5998–6008.
- Zhengli Zhao, Dheeru Dua, and Sameer Singh. 2018. Generating natural adversarial examples. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

## A Supplemental Material

### A.1 Generating OOV Replacements with Internal Character Swaps

We use the following snippet to produce an OOV word from an existing word:

```
1 def make_oov(  
2     word,  
3     vocab,  
4     max_scrambling,  
5 ):  
6     """Modify a word to make it OOV  
7     (while keeping the meaning)"""  
8     # If the word has >3 letters  
9     # try scrambling them  
10    L = len(word)  
11    if L > 3:  
12        # For a fixed number of steps  
13        for _ in range(max_scrambling):  
14            # Swap two adjacent letters  
15            # in the middle of the word  
16            pos = random.randint(1, L - 3)  
17            word = word[:pos]  
18            word += word[pos+1] + word[pos]  
19            word += word[pos+2:]  
20            # If we got an OOV already just  
21            # return it  
22            if word not in vocab:  
23                return word  
24    # If nothing worked, or the word is  
25    # too short for scrambling, just  
26    # repeat the last letter ad nauseam  
27    char = word[-1]  
28    while word in vocab:  
29        word = word + char  
30    return word
```

### A.2 IWSLT2016 Dataset

See table 7 for statistics on the size of the IWSLT2016 corpus used in our experiments.

	#train	#valid	#test
fr-en	220.4k	6,824	2,213
de-en	196.9k	11,825	2,213
cs-en	114.4k	5,716	2,213

Table 7: IWSLT2016 data statistics.

### A.3 Breakdown of Correlation with Human Judgement

We provide a breakdown of the correlation coefficients of automatic metrics with human judgment for source-side meaning-preservation, both in terms of number of perturbed words (Table 8) and constraint (Table 9). While those coefficients are computed on a much smaller sample size, and their differences are not all statistically significant with  $p < 0.01$ , they exhibit the same trend as the results from Table 3 (BLEU < METEOR < chrF).

# edits	BLEU	METEOR	chrF
1	0.351	0.352	<b>0.486*</b>
2	0.403	0.424	<b>0.588*</b>
3	0.334	0.393	<b>0.560*</b>

Table 8: Correlation of automatic metrics to human judgment of semantic similarity between original and adversarial source sentences, broken down by number of perturbed words. “\*” indicates that the correlation is significantly better than the next-best one.

Constraint	BLEU	METEOR	chrF
Unconstrained	0.274	0.572	<b>0.599</b>
CharSwap	0.274	0.319	<b>0.383</b>
kNN	0.534	0.584	<b>0.606</b>

Table 9: Correlation of automatic metrics to human judgment of semantic similarity between original and adversarial source sentences, broken down by type of constraint on the perturbation. “\*” indicates that the correlation is significantly better than the next-best one.

In particular Table 8 shows that the good correlation of chrF with human judgment is not only due to the ability to distinguish between different number of edits.