On the Robustness of Self-Attentive Models

Yu-Lun Hsieh[1,2], **Minhao Cheng**[3], **Da-Cheng Juan**[4], **Wei Wei**[4],
Wen-Lian Hsu[1,5], **Cho-Jui Hsieh**[3,4]
[1]SNHCC, TIGP, Academia Sinica, Taiwan
[2]National Chengchi University, Taiwan
[3]University of California, Los Angeles, USA
[4]Google Research, USA
[5]PAIR Labs, Ministry of Science and Technology, Taiwan
morphe@iis.sinica.edu.tw, mhcheng@cs.ucla.edu, x@dacheng.info, wewei@google.com,
hsu@iis.sinica.edu.tw, chohsieh@cs.ucla.edu

Abstract

This work examines the robustness of self-attentive neural networks against adversarial input perturbations. Specifically, we investigate the attention and feature extraction mechanisms of state-of-the-art recurrent neural networks and self-attentive architectures for sentiment analysis, entailment and machine translation under adversarial attacks. We also propose a novel attack algorithm for generating more natural adversarial examples that could mislead neural models but not humans. Experimental results show that, compared to recurrent neural models, self-attentive models are more robust against adversarial perturbation. In addition, we provide theoretical explanations for their superior robustness to support our claims.

1 Introduction

Self-attentive neural models have recently become a prominent component that achieves state-of-the-art performances on many natural language processing (NLP) tasks such as text classification and machine translation (MT). This type of models, including Transformer (Vaswani et al., 2017) and "Bidirectional Encoder Representations from Transformers," shortened as *BERT* (Devlin et al., 2019), rely on the attention mechanism (Luong et al., 2015) to learn a context-dependent representation; compared to recurrent neural networks (RNN), these self-attention-based models have faster encoding speed and the capacity of modeling a wider context. Particularly, BERT is recently proposed to extend the directionality of the Transformer model, and "pre-trained" using multiple objectives to strengthen its encoding capability. Then, this pre-trained model can be fine-tuned for various downstream tasks. BERT achieves state-of-the-art performance on several NLP tasks including classification and sequence-to-sequence problems, often outperforming task-specific feature engineering or model architecture; therefore, BERT is poised to be a key component in almost every neural model for NLP tasks.

Despite the superior performance, it remains unclear whether the self-attentive structure deployed by Transformer or BERT is robust to adversarial attacks compared with other neural networks. Adversarial attack refers to applying a small perturbation on the model input to craft an adversarial example, ideally imperceptible by humans, and cause the model to make an incorrect prediction (Goodfellow et al., 2015). Unlike computer vision models, generating an effective, textual adversarial example that misleads a model but can go unnoticed by humans is a challenging and thriving research problem (Alzantot et al., 2018). Therefore, the goal of this paper is to answer the following questions: "*Are self-attentive models more robust to adversarial examples compared with recurrent models? If so, why?*" "*Do attention scores expose vulnerability in these self-attentive models?*"

This work verifies the robustness of self-attentive models through performing adversarial attacks and analyzing their effects on the model prediction. In addition, we investigate the feasibility of utilizing the context-dependent embeddings in these models to maximize semantic similarity between real and adversarial sentences. We conduct experiments on two mainstream self-attentive models: (a) Transformer for neural machine translation, and (b) BERT for sentiment and entailment classification. To the best of our knowledge, this paper brings the following contributions.

- We propose novel algorithms to generate more natural adversarial examples that both preserve the semantics and mislead the classifiers.
- We conduct comprehensive experiments to

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 1520–1529
Florence, Italy, July 28 - August 2, 2019.

examine the robustness of RNN, Transformer, and BERT. Our results show that both self-attentive models, whether pre-trained or not, are more robust than recurrent models.
- We provide theoretical explanations to support the statement that self-attentive structures are more robust to small adversarial perturbations.

2 Target Neural Models

This section describes the target neural architectures, LSTM and self-attentive models, and how to adapt these models for the downstream tasks: sentiment analysis, entailment and translation.

2.1 LSTM

For classification tasks including sentiment analysis and entailment detection, we use a Bidirectional LSTM with an attention (Hochreiter and Schmidhuber, 1997; Bahdanau et al., 2014) layer as the sentence encoder, and a fully connected layer for classification problems. For machine translation, we employ a common seq2seq model (Sutskever et al., 2014), in which both the encoder and decoder are a 2-layer stacked BiLSTM with 512 hidden units.

2.2 Self-Attentive Models

Self-attentive models are further distinguished into BERT and Transformers. The classification problems adopt the BERT model with an identical setup to the original paper (Devlin et al., 2019), in which BERT is used as an encoder that represents a sentence as a vector. This vector is then used by a fully connected neural network for classification. Note that models are tuned separately for each task. We also experiment with a smaller BERT model without pre-training, denoted as $\text{BERT}_{\text{NOPT}}$, in order to isolate the impact of pre-training. Due to the limited size of the training data, we only incorporate three layers of self-attention in the smaller model.

To the best of our knowledge, there is no prior work that uses pre-trained BERT for machine translation. Thus, the Transformer model is employed for neural machine translation task.

3 Attack Methods

In this section, we provide five methods to generate adversarial examples (or called "attacks"). The goal of an attack is to find and replace **one** word in the original input sentence, turning the output label (or sequence) from the model to be incorrect. The first method is based on random word replacement, which serves as the baseline. The second (list-based) and third (greedy) methods are adapted from prior arts. The fourth (constrained greedy) and fifth (attention-based) are proposed by us. We also describe the evaluation metrics.

3.1 Random Attack

This attack randomly replaces one word in the input sentence with another word from the vocabulary. We repeat this process by 10^5 times and calculate the average as the final performance. This baseline is denoted as **RANDOM**.

3.2 List-based Attack

The second method is recently proposed by Alzantot et al. (2018), denoted as **LIST**. LIST employs a list of semantically similar words (*i.e.*, *synonyms*), and manages to replace a word in the input sentence with another from the list to construct adversarial examples. In other words, the list is used to replace a word with one of its synonyms; this process is repeated for every word in the input sentence until the target model makes an incorrect prediction. That is, for every sentence, we start by replacing the first word with its synonyms, each forming a new adversarial example. If none of these successfully misleads the model, we move to the next word (and the first word remains unchanged), and repeat this process until either the attack succeeds or all words have been tried.

3.3 Greedy Select + Greedy Replace

The third method (denoted as **GS-GR**) greedily searches for the weak spot of the input sentence (Yang et al., 2018) by replacing each word, one at a time, with a "padding" (a zero-valued vector) and examining the changes of output probability. After determining the weak spot, GS-GR then replaces that word with a randomly selected word in the vocabulary to form an attack. This process is repeated until the attack succeeds or all words in the vocabulary are exhausted.

3.4 Greedy Select + Embedding Constraint

Although the GS-GR method potentially achieves a high success rate, the adversarial examples formed by GS-GR are usually unnatural; sometimes GS-GR completely changes the semantics of

the original sentence by replacing the most important word with its antonym, for example: changing "this is a **good** restaurant" into "this is a **bad** restaurant." This cannot be treated as a successful attack, since humans will notice the change and agree with the model's output. This is because GS-GR only considers the classification loss when finding the replacement word, and largely ignore the actual semantics of the input sentence.

To resolve this issue, we propose to add a constraint on sentence-level (not word-level) embedding: the attack must find a word with the minimum L1 distance between two embeddings (from the sentences before and after the word change) as the replacement. This distance constraint requires a replacement word not to alter the sentence-level semantics too much. This method is denoted as **GS-EC**. In the experimental results, we show that the GS-EC method achieves a similar success rate as GS-GR in misleading the model, while being able to generate more natural and semantically-consistent adversarial sentences.

3.5 Attention-based Select

We conjecture that self-attentive models rely heavily on attention scores, and changing the word with the highest or lowest attention score could substantially undermine the model's prediction. Therefore, this attack method exploits and also investigates the attention scores as a potential source of vulnerability. This method first obtains the attention scores and then identifies a target word that has the highest or lowest score. Target word is then replaced by a random word in the vocabulary, and this process is repeated until the model is misled by the generated adversarial example. These methods are denoted as **AS_{MIN}-GR** that replaces the word with the lowest score, and **AS_{MAX}-GR** with the highest score.

Furthermore, the constraint on the embedding distance can also be imposed here for finding semantically similar adversarial examples; these methods are referred as **AS_{MIN}-EC** and **AS_{MAX}-EC**, respectively. As a pilot study, we examine the attention scores on the first and last layers of the BERT model for understanding the model's behavior under attacks.

3.6 Evaluation Criteria

We evaluate the robustness of the classification models (for sentiment analysis and entailment) by the following three criteria: (a) the success rate of the attacks misleading the model, (b) readability, and (c) human accuracy. Both readability and human accuracy are evaluated qualitatively by human raters. Readability measures the relative naturalness of the adversarial examples generated by different attack methods. For example, if 100 raters determine that the adversary generated by method A is more readable than method B, and 40 raters think otherwise, the relative readability scores of methods A and B will be 1 and 0.4, respectively. And human accuracy is the percentage that human judgment of these examples remains identical to the ground-truth label. In order to evaluate the models and at the same time keep reasonable execution time, we randomly select 100 samples from the test set that all models answer correctly to perform attacks. For the experiments on machine translation task, we evaluate the attack success rate and BLEU scores (Papineni et al., 2002) for 200 sentence pairs in the WMT 17 Task (Bojar et al., 2017).

4 Experiment I: Sentiment Analysis

We first evaluate the robustness of LSTM, BERT, and $\text{BERT}_{\text{NOPT}}$ on binary sentiment analysis using the Yelp dataset (Zhang et al., 2015). Models under attack have accuracies of 93.7%, 87.3% and 90.7% for fine-tuned BERT model, $\text{BERT}_{\text{NOPT}}$ and LSTM, respectively, on the test set. Note that for attention-based attacks (*i.e.*, AS_{MIN}-GR, AS_{MAX}-GR, AS_{MIN}-EC, and AS_{MAX}-EC), the average of the first (*i.e.*, the one that is closest to the model input) attention layer from all 12 heads in BERT and $\text{BERT}_{\text{NOPT}}$ are used for our attacks.[1]

4.1 Results

To illustrate how adversarial attacks work, Fig. 1 shows the results from AS_{MAX}-EC and AS_{MIN}-EC methods that select a word to change based on the attention scores of the original sentence. A comprehensive quantitative comparison can be found in Table 1, from which we make the following observations:

- Greedy-based attacks consistently achieve higher successful rate than other attacks. The proposed GS-EC method can achieve almost identical success rates with GS-GR while restricting the search space based on the embedding distances. We will further show that

[1] As an alternative, we tested using the last layer during AS_{MAX}-ECattack. However, experimental results exhibit a $< 10\%$ success rate.

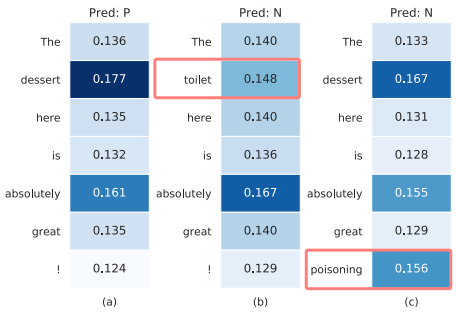


Figure 1: Illustrations of attention scores of (a) the original input, (b) AS_{MIN}-EC, and (c) AS_{MAX}-EC attacks. The attention-based methods select words based on the maximum or minimum attention, which is annotated by red boxes. Both of them reversed the predicted sentiment of the sentence from positive to negative.

Attack Method	Model LSTM	BERT	$BERT_{NOPT}$
RANDOM	1.1%	0.8%	1%
LIST	27%	6%	15%
AS_{MIN}-GR	16%	11%	32%
AS_{MAX}-GR	62%	17%	35%
AS_{MIN}-EC	16%	10%	32%
AS_{MAX}-EC	62%	17%	35%
Best attention attack(A_*)	62%	17%	35%
GS-GR	**79%**	**52%**	**53%**
GS-EC	*78%*	*50%*	**53%**

Table 1: Success rates of attack methods across models for sentiment analysis. Bold numbers indicate the highest attack rate in a column.

GS-EC leads to higher quality adversarial examples in Section 4.2.

- We found that using attention, especially **AS_{MAX}** methods, can easily break the LSTM model. However, the same vulnerability does not exist in BERT or $BERT_{NOPT}$ models. Since different types of attention-based attacks are suitable for different models, we summarize the best attention-based attack performance as A_* in the table, which takes the maximum over four different types of attention-based attacks.
- Self-attentive models (BERT and $BERT_{NOPT}$) consistently lead to lower attack successful rates compared with the LSTM model, under RANDOM, LIST, attention-based attacks and greedy-based attacks.

We demonstrate the robustness of BERT model under GS-EC attack in Fig 2. We can see that, GS-EC caused a substantial shift in the LSTM's attention map while that of BERT remain stable.

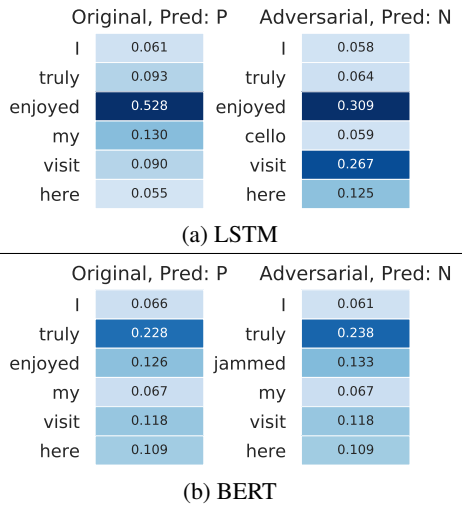


Figure 2: Attention scores in (a) LSTM and (b) BERT models under GS-EC attacks. Although GS-EC successfully flips the predicted sentiment for both models from positive to negative, the attention scores remain stable for BERT model. The LSTM model, however, suffers from a large shift in attention distribution.

Method	Sentence
GS-GR	Pizzeria Bianco was a ~~*such*~~ **never** a nice treat that was [...]
GS-EC	Pizzeria Bianco was a ~~*such*~~ **ostensibly** a nice treat that was [...]
GS-GR	The desserts here are absolutely ~~*great*~~ **0** ! [...]
GS-EC	The desserts here are absolutely ~~*great*~~ **soluble** ! [...]

Table 2: Adversarial examples for the BERT sentiment analysis model generated by GS-GR and GS-EC methods. Both attacks caused the prediction of the model to change. Note here that GS-EC model selects a word that preserves local coherency due to the similarity constraints. GS-GR model, on the contrary, finds a word that is less coherent with the context.

4.2 Quality of Adversarial Examples

We conduct experiments to assess the naturalness of adversarial examples. First, Table 2 compares the quality of the results generated by GS-GR and GS-EC attacks on a BERT model. Here we see that constraints imposed by GS-EC make it superior than GS-GR in terms of retrieving words that are coherent with the context.

Furthermore, we organize a large-scale human evaluation on Amazon Mechanical Turk regarding the qualities of adversarial examples generated by different methods. Each sample is voted by 3 turkers. Recall that we define "Readability" and "Human accuracy" in Section 3.6. Readability is regarded as the relative naturalness of the adver-

sarial examples, normalized to the maximum between the compared methods. The human accuracy metric is the percentage of human responses that matches the true label. Table 3 is a comparison of LSTM and BERT models using the GS-EC attack. It shows that the distance in embeddings space of BERT can better reflect semantic similarity and contribute to more natural adversarial examples. And, in Table 4, we compare using GS-GR and GS-EC method on BERT model. Again, we see that the GS-EC method, which restricts the distance between sentence embeddings of original and adversarial inputs, can produce superior adversaries.

Model	Readability	Human Accuracy
LSTM	0.6	52.1%
BERT	**1.0**	**68.8%**

Table 3: Comparison of LSTM and BERT models under human evaluations against GS-EC attack. **Readability** is a relative quality score between models, and **Human Accuracy** is the percentage that human raters correctly identify the adversarial examples.

Method	Readability	Human Accuracy
GS-GR	0.55	64.6%
GS-EC	**1.0**	**68.8%**

Table 4: Comparison of GS-GR and GS-EC attacks on BERT model for sentiment analysis. **Readability** is a relative quality score between attack methods, and **Human Accuracy** is the percentage that human raters correctly identify the sentiment of adversarial examples.

5 Experiment II: Textual Entailment

We conduct evaluations on MultiNLI (Williams et al., 2018) dataset for textual entailment with approaches similar to the ones in the last section. MultiNLI is one of the many datasets that see major improvements by BERT. The BERT model is trained to achieve 83.5% accuracy and LSTM 76%. $\text{BERT}_{\text{NOPT}}$ is excluded from this experiment since it cannot reach a satisfactory accuracy.

5.1 Results

Results from entailment models fall into the same pattern as those from sentiment analysis, which is listed in Table 5. Our findings are summarized as follows:

- The entailment task is more difficult than single-sentence classification, as evidenced by the higher success rates of attacks among all models and attacks.
- The greedy-based attacks consistently achieve higher success rates.
- AS_{MAX} methods continue to be superior than AS_{MIN}, although the difference here is not as drastic as in the previous experiment.
- BERT model remains more robust compared with LSTM.

	Model	
Attack Method	**LSTM**	**BERT**
RANDOM	17.8%	9.2%
LIST	63%	56%
AS_{MIN}-GR	57%	53%
AS_{MAX}-GR	78%	54%
AS_{MIN}-EC	55%	52%
AS_{MAX}-EC	78%	51%
Best attention attack(A_*)	78%	54%
GS-GR	**95%**	**75%**
GS-EC	**95%**	**75%**

Table 5: Success rate of different attack methods on LSTM and BERT for the MultiNLI development set.

5.2 Quality of Adversarial Examples

Samples illustrated in Table 6 show that the GS-EC method can find more coherent words for the attack, as opposed to GS-GR. For instance, changing the word "great" to "vast" can cause the model to misjudge the entailment relation in the second example. Unfortunately due to budget constraints, we did not conduct large scale human experiments on this dataset.

6 Experiment III: Machine Translation

We implement LSTM and Transformer machine translation models using OpenNMT-py[2]. Specifically, for the LSTM model, we train it with 453 thousand pairs from the Europarl corpus of German-English WMT 15 Task[3], common crawl, and news-commentary. The LSTM model is a two-layer bidirectional LSTM with 512 hidden units together with a attention layer. We use the default hyper-parameters, and reproduce the performance reported by Ha et al. (2016). For the Transformer, we use a public pre-trained model with 6 self-attention layers provided by OpenNMT-py that reproduces the performance reported by Vaswani et al. (2017).

[2]https://github.com/OpenNMT/OpenNMT-py
[3]http://www.statmt.org/wmt15/translation-task.html

Label	Sentence 1	Sentence 2
Contradiction →Neutral	No, I don't know.	(Original) Yes , I *know*. (GS-GR) Yes, I **0**. (GS-EC) Yes, I **renovated**.
Neutral →Contradiction	That's it. The girl looked at him, then passed her hand across her forehead.	(Original) The girl looked at him with *great* interest. (GS-GR) The girl looked at him with **!** interest. (GS-EC) The girl looked at him with **vast** interest.
Entailment →Neutral	(Original) Workers are also represented in civil rights and *retaliation* claims. (GS-GR) Workers are also represented in civil rights and **?** claims. (GS-EC) Workers are also represented in civil rights and **targets** claims.	Some workers are represented in civil rights and retaliation claims.

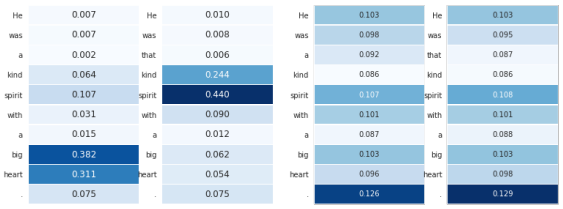
Table 6: Adversarial examples generated by GS and GS-EC attacks for BERT entailment classifier.

Unlike the classification tasks, in machine translation the attack goal is harder to define. We chose to evaluate the robustness under two types of attacks. In the first type of "targeted keyword attack" discussed in (Cheng et al., 2018), we attempt to generate an adversarial input sequence such that a specific keyword appears in the output sequence within the threshold Δ of number of word changes we allowed. Empirically, we set $\Delta = 3$ in these experiments and adopt the most successful attack, GS-EC, to this case. For the second type of untargeted attack, we consider perturbing the input to degrade the BLUE score of output sequences with respect to the ground-truths. For doing this, we conduct a typo-based attack (Belinkov and Bisk, 2018). Specifically, we randomly select one word in each sentence and change it to a typo predefined in a common typo list. This can be viewed as an extension of LIST attack to the translation task.

6.1 Results

For the targeted keyword attack, the success rates on both models are reported in Table 7. First, we notice that the success rate of the attacks are below 30%, presumably because translation is substantially more complex compared with the aforementioned text classification tasks. Nevertheless, the attacks on the Transformer model is significantly less successful than the LSTM-based one.

For the typo-based attack, the BLUE scores before/after the attack are reported in Table 8. We observe that the Transformer-based model always achieves a higher BLEU score over LSTM-based model, *i.e.*, have a better translation performance whether the sentences contain typos or not. We conclude that Transformer-based model exhibits a greater robustness over LSTM-based model in the case of machine translation. This is consistent with our findings in the previous experiments on sentiment and entailment classification problems.



(a) LSTM (b) Transformer

Figure 3: Compare attention scores of the original versus adversarial inputs for LSTM and Transformer models for machine translation.

In addition, we present some successful adversarial examples in Table 9, and see that the greedy attack can indeed generate natural examples for both models.

Attack Method	LSTM	Transformer
GS-EC	27.5%	10.5%

Table 7: Targeted attack success rate with GS-EC in translation tasks.

Model	Original	Adversarial
LSTM	25.10	13.44
Transformer	**34.90**	**26.02**

Table 8: BLEU scores using typo-based attack on LSTM and Transformer translation models.

7 Theoretical Analysis

All the above experiments conclude that a self-attentive model exhibits higher robustness compared to a recurrent one. This is somewhat counter-intuitive—at the first glance one may assume that the self-attention layer is not robust

LSTM	Original input	There is a fundamental philosophical reason for the differences between Donald Trump's and Hillary Clinton's [...]
	Adv input	There is a fundamental philosophical **r** for the differences between Donald Trump's and Hillary Clinton's [...]
	Original output	Es gibt einen grundlegenden philosophischen Grund für die Unterschiede zwischen Donald Trump und Hillary Clinton s
	Adv output	Es gibt eine grundlegende philosophischer **Art** , wie Unterschied e zwischen Donald Trump und Hillary Clinton s
TF	Original input	And in this vein , he passed the prize money of 2 5,000 euros on straight away
	Adv input	And **as** this vein , he passed the prize money of 2 5,000 euros on straight away
	Original output	Und in diesem Sinne hat er sofort das Preis geld von 2 5.000 Euro über wiesen
	Adv output	Und als diese **Art** , ging er sofort das Preis geld von 2 5.000 Euro weiter

Table 9: Adversarial examples for LSTM and Transformer (shortened as **TF**) models with the target keyword "Art." in the output.

since perturbation in one word can affect all the attention scores. In this section, we provide some explanation regarding this phenomenon by studying how error propagates through the self-attention architecture. We show that the perturbation of one input embedding can in fact only have sparse affect to the attention scores when the input embedding are scattered enough in the space.

Sensitivity of Self-Attention Layers : First, we consider the simple case of one self-attention layer with a single head. Assume a sentence has n input words and each word is represented by a d-dimensional embedding vectors, denoted by $x_1, \ldots, x_n \in R^d$. We use $W^Q, W^K, W^V \in R^{d\times k}$ to denote the query, key and value transformations. The contribution of each element j to i is then computed by

$$s_{ij} = x_i^T W^Q (W^K)^T x_j,$$

and then the i-th embedding at the next layer is obtained by

$$z_i = \sum_j \frac{e^{s_{ij}}}{\sum_k e^{s_{ik}}} (W^V x_j),$$

Sometimes z_i is fed into another linear layer to obtain the embeddings. Now, consider that a small perturbation is added to a particular index $\bar{j}$, such that $x_{\bar{j}}$ is changed to $x_{\bar{j}} + \Delta x$ while all the other $\{x_j \mid j \neq \bar{j}\}$ remain unchanged. We then study how much this perturbation will affect $\{z_i\}_{i\in[n]}$. For a particular i ($\neq j$), the s_{ij} is only changed by one term since

$$s'_{ij} = \begin{cases} s_{ij} & \text{if } j \neq \bar{j} \\ s_{ij} + x_i^T W^Q (W^K)^T \Delta x & \text{if } j = \bar{j} \end{cases} \quad (1)$$

where we use s'_{ij} to denote the value after the perturbation. Therefore, with the perturbed input, each set of $\{s_{ij}\}_{j=1}^n$ will only have one term being changed. Furthermore, the changed term in equation 1 is the inner product between x_i and a fixed vector $W^Q(W^K)^T\Delta x$; although this could be large for some particular x_i in the similar direction of $W^Q(W^K)^T\Delta x$, if the embeddings $\{x_i\}_{i=1}^n$ are scattered enough over the space, the inner products cannot be large for all $\{x_i\}_{i=1}^n$. Therefore, the change to the next layer will be sparse. For instance, we can prove the sparsity under some distributional assumptions on $\{x_i\}$:

Theorem 1. *Assume $\|\Delta x\| \leq \delta$ and $\{x_i\}_{i=1}^n$ are d-dimensional vectors uniformly distributed on the unit sphere, then $E[|s'_{i\bar{j}} - s_{i\bar{j}}|] \leq \frac{C\delta}{\sqrt{d}}$ with $C = \|W^Q\|\|W^K\|$ and $P(|s'_{i\bar{j}} - s_{i\bar{j}}| \geq \epsilon) \leq \frac{C\delta}{\epsilon\sqrt{d}}$.*

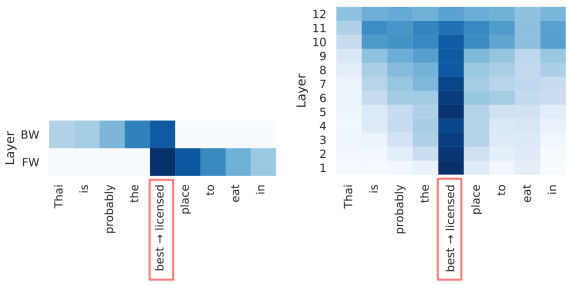
Proof. The value $E[s'_{i\bar{j}} - s_{i\bar{j}}] = E[x_i^T z]$ where $z = W^Q(W^K)^T\Delta x$ is a fixed vector, and it is easy to derive $\|z\| \leq \|W^Q\|\|W^K\|\delta$. To bound this expectation, we first try to bound $a_1 = E[x_i^T e_1]$ where $e_1 = [1, 0, \ldots, 0]$. Due to the rotation invariance we can obtain $a_1 = \cdots = a_d$ and $\sum_i a_i^2 = 1$, so $|a_1| = \frac{1}{\sqrt{d}}$. This implies $E[x_i^T z] \leq \frac{C\delta}{\sqrt{d}}$. Using Markov inequality, we can then find the probability results. □

Therefore, as the norm of W^Q, W^K are not too large (usually regularized by L2 during training) and the dimension d is large enough, there will be a significant amount of i such that s_{ij} is perturbed negligibly.

In contrast, embeddings from RNN-based models are relatively more sensitive to perturbation of one word, as shown below. Similar to the previous case, we assume a sequence $x_1, \ldots, x_n$, and a word $x_{\bar{j}}$ is perturbed by Δx. For the vanilla RNN model, the embeddings are sequentially computed as $z_i = \sigma(Ax_i + Bz_{i-1})$. If $x_{\bar{j}}$ is perturbed, then all the $\{z_i\}_{i=\bar{j}}^n$ will be altered. Therefore, the at-

tacker can more easily influence all the embeddings.

As an illustration of the proposed theory, we plot a comparison of the degree of embeddings variation from two models after changing one word in Fig. 4. We observe that, for self-attentive models, the distribution of change on embeddings is sparse after going through the first self-attention layer (layer 1) and then gradually propagate to the whole sequence when passing through more layers. In contrast, the embeddings from LSTM exhibit a denser pattern. To further validate our analysis, we calculate the ratio of the L2 norms of embeddings variation. Specifically, let $\boldsymbol{z}$ and $\boldsymbol{z}_{\text{adv}}$ denote the embeddings of the original sentence and adversarial input, respectively. We represent relative embedding variation $\mathcal{R}_e = \|\boldsymbol{z} - \boldsymbol{z}_{\text{adv}}\|/\|\boldsymbol{z}\|$. For the GS-EC attack in the sentiment analysis task, embeddings from the LSTM model has an average $\mathcal{R}_e$ of 0.83 whereas for the BERT model it is 0.56 under the same attack by changing one word. This supports our claim that the impact of an adversarial example is more severe on the LSTM model than BERT, which presumably plays an important role in the robustness of self-attentive models.



(a) LSTM (b) BERT

Figure 4: Comparison of L2 norm of embedding variations after changing one word (marked by red box) in the input to (a) LSTM (b) BERT.

8 Related Work

Robustness of neural network models has been a prominent research topic since Szegedy et al. (2013) discovered that CNN-based image classification models are vulnerable to adversarial examples. However, attempts to examine the robustness of NLP models are relatively few and far between. Previous work on attacking neural NLP models include using Fast Gradient Sign Method (Goodfellow et al., 2015) to perturb the embedding of RNN-based classifiers (Papernot et al., 2016; Liang et al., 2017), but they have difficulties mapping from continuous embedding space to discrete input space. Ebrahimi et al. (2018) propose the 'HotFilp' method that replaces the word or character with the largest difference in the Jacobian matrix. Li et al. (2016) employ reinforcement learning to find the optimal words to delete in order to fool the classifier. More recently, Yang et al. (2018) propose a greedy method to construct adversarial examples by solving a discrete optimization problem. They show superior performance than previous work in terms of attack success rate, but the greedy edits usually degrade the readability or significantly change the semantics. Zhao et al. (2018) utilize generative adversarial networks (GAN) to generate adversarial attacks against black-box models for applications including image classification, textual entailment, and machine translation. Alzantot et al. (2018) propose to use a pre-compiled list of semantically similar words to alleviate this issue, but leads to lower successful rate as shown in our experiments. We thus include the latest greedy and list-based approaches in our comparisons.

In addition, the concept of adversarial attacks has also been explored in more complex NLP tasks. For example, Jia and Liang (2017) attempt to craft adversarial input to a question answering system by inserting irrelevant sentences at the end of a paragraph. Cheng et al. (2018) develop an algorithm for attacking seq2seq models with specific constraints on the content of the adversarial examples. Belinkov and Bisk (2018) compare typos and artificial noise as adversarial input to machine translation models. Also, Iyyer et al. (2018) propose a paraphrase generator model learned from back-translation data to generate legitimate paraphrases of a sentence as adversaries. However, the semantic similarity is not guaranteed. In terms of comparisons between LSTM and Transformers, Tang et al. (2018) show that multi-headed attention is a critical factor in Transformer when learning long distance linguistic relations.

This work is unique in a number of aspects. First, we examine the robustness of uni- and bi-directional self-attentive model as compared to recurrent neural networks. And, we devise novel attack methods that take advantage of the embedding distance to maximize semantic similarity between real and adversarial examples. Last but not least, we provide detail observations of the inter-

nal variations of different models under attack and theoretical analysis regarding their levels of robustness.

9 Conclusions

We show that self-attentive models are more robust to adversarial attacks than recurrent networks under small input perturbations on three NLP tasks, *i.e.*, sentiment analysis, entailment, and translation. We provide theoretical explanations regarding why the self-attention structure leads to better robustness, in addition to illustrative examples that visualize the model's internal variations. Future work includes developing a adversarial training scheme as well as devising a more robust architecture based on our findings.

Acknowledgements

We are grateful for the insightful comments from anonymous reviewers. This work is supported by the Ministry of Science and Technology of Taiwan under grant numbers 107-2917-I-004-001, 108-2634-F-001-005. The author Yu-Lun Hsieh wishes to acknowledge, with thanks, the Taiwan International Graduate Program (TIGP) of Academia Sinica for financial support towards attending this conference. We also acknowledge the support from NSF via IIS1719097, Intel and Google Cloud.

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. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

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

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 conference on machine translation (wmt17). In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 169–214, Copenhagen, Denmark. Association for Computational Linguistics.

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. *CoRR*, abs/1803.01128.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. 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)*, volume 2, pages 31–36.

Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*.

Thanh-Le Ha, Jan Niehues, and Alexander Waibel. 2016. Toward multilingual neural machine translation with universal encoder and decoder. *arXiv preprint arXiv:1611.04798*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Lstm can solve hard long time lag problems. In *Advances in neural information processing systems*, pages 473–479.

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)*, volume 1, 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.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. Understanding neural networks through representation erasure. *arXiv preprint arXiv:1612.08220*.

Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. 2017. Deep text classification can be fooled. *arXiv preprint arXiv:1704.08006*.

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.

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 the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Gongbo Tang, Mathias Müller, Annette Rios, and Rico Sennrich. 2018. Why self-attention? a targeted evaluation of neural machine translation architectures. In *Conference on Empirical Methods in Natural Language Processing*, pages 4263–4272.

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 *Advances in Neural Information Processing Systems*, pages 5998–6008.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. 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 1112–1122. Association for Computational Linguistics.

Puyudi Yang, Jianbo Chen, Cho-Jui Hsieh, Jane-Ling Wang, and Michael I Jordan. 2018. Greedy attack and gumbel attack: Generating adversarial examples for discrete data. *arXiv preprint arXiv:1805.12316*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 649–657. Curran Associates, Inc.

Zhengli Zhao, Dheeru Dua, and Sameer Singh. 2018. Generating natural adversarial examples. In *International Conference on Learning Representations*.