Effective Adversarial Regularization for Neural Machine Translation

Motoki Sato[1], **Jun Suzuki**[2,3], **Shun Kiyono**[3,2]
[1]Preferred Networks, Inc., [2]Tohoku University,
[3]RIKEN Center for Advanced Intelligence Project
sato@preferred.jp, jun.suzuki@ecei.tohoku.ac.jp, shun.kiyono@riken.jp

Abstract

A regularization technique based on adversarial perturbation, which was initially developed in the field of image processing, has been successfully applied to text classification tasks and has yielded attractive improvements. We aim to further leverage this promising methodology into more sophisticated and critical neural models in the natural language processing field, i.e., neural machine translation (NMT) models. However, it is not trivial to apply this methodology to such models. Thus, this paper investigates the effectiveness of several possible configurations of applying the adversarial perturbation and reveals that the adversarial regularization technique can significantly and consistently improve the performance of widely used NMT models, such as LSTM-based and Transformer-based models.[1]

1 Introduction

The existence of (small) perturbations that induce a critical prediction error in machine learning models was first discovered and discussed in the field of image processing (Szegedy et al., 2014). Such perturbed inputs are often referred to as *adversarial examples* in the literature. Subsequently, Goodfellow et al. (2015) proposed a learning framework that simultaneously leverages adversarial examples as additional training data for reducing the prediction errors. This learning framework is referred to as *adversarial training*.

In the field of natural language processing (NLP), the input is a sequence of discrete symbols, such as words or sentences. Since it is unreasonable to add a small perturbation to the symbols, applying the idea of adversarial training to NLP tasks has been recognized as a challenging problem. Recently, Miyato et al. (2017) overcame this problem and reported excellent performance improvements on multiple benchmark datasets of text classification task. The key idea of their success is to apply adversarial perturbations into the input embedding layer instead of the inputs themselves as used in image processing tasks. An important implication of their study is that their method can be interpreted as a regularization method, and thus, they do not focus on generating adversarial examples. We refer to this regularization technique as *adversarial regularization*.

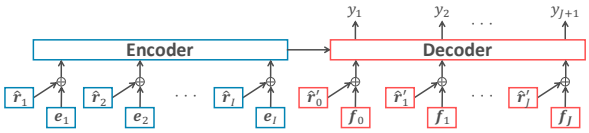

Figure 1: An intuitive sketch that explains how we add adversarial perturbations to a typical NMT model structure for adversarial regularization. The definitions of $\boldsymbol{e}_i$ and $\boldsymbol{f}_j$ can be found in Eq. 2. Moreover, those of $\hat{\boldsymbol{r}}_i$ and $\hat{\boldsymbol{r}}'_j$ are in Eq. 8 and 13, respectively.

We aim to further leverage this promising methodology into more sophisticated and critical neural models, i.e., neural machine translation (NMT) models, since NMT models recently play one of the central roles in the NLP research community; NMT models have been widely utilized for not only NMT but also many other NLP tasks, such as text summarization (Rush et al., 2015; Chopra et al., 2016), grammatical error correction (Ji et al., 2017), dialog generation (Shang et al., 2015), and parsing (Vinyals et al., 2015; Suzuki et al., 2018). Unfortunately, this application is not fully trivial since we potentially have several configurations for applying adversarial perturbations into NMT models (see details in Section 5). Figure 1 illustrates the model architecture of NMT models with adversarial perturbation.

Therefore, the goal of this paper is to re-

[1]Our code for replicating the experiments in this paper is available at the following URL: https://github.com/pfnet-research/vat_nmt

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 204–210
Florence, Italy, July 28 - August 2, 2019.

veal the effectiveness of the adversarial regularization in NMT models and encourage researchers/developers to apply the adversarial regularization as a common technique for further improving the performance of their NMT models. We investigate the effectiveness of several possible configurations that can significantly and consistently improve the performance of typical baseline NMT models, such as LSTM-based and Transformer-based models,

2 Related Work

Several studies have recently applied adversarial training to NLP tasks, e.g., (Jia and Liang, 2017; Belinkov and Bisk, 2018; Hosseini et al., 2017; Samanta and Mehta, 2017; Miyato et al., 2017; Sato et al., 2018). For example, Belinkov and Bisk (2018); Hosseini et al. (2017) proposed methods that generate input sentences with random character swaps. They utilized the generated (input) sentences as additional training data. However, the main focus of these methods is the incorporation of *adversarial examples* in the training phase, which is orthogonal to our attention, *adversarial regularization*, as described in Section 1.

Clark et al. (2018) used virtual adversarial training (VAT), which is a semi-supervised extension of the adversarial regularization technique originally proposed in Miyato et al. (2016), in their experiments to compare the results with those of their proposed method. Therefore, the focus of the neural models differs from this paper. Namely, they focused on sequential labeling, whereas we discuss NMT models.

In parallel to our work, Wang et al. (2019) also investigated the effectiveness of the adversarial regularization technique in neural language modeling and NMT. They also demonstrated the impacts of the adversarial regularization technique in NMT models. We investigate the effectiveness of the several practical configurations that have not been examined in their paper, such as the combinations with VAT and back-translation.

3 Neural Machine Translation Model

Model Definition In general, an NMT model receives a sentence as input and returns a corresponding (translated) sentence as output. Let $\mathcal{V}_{\mathrm{s}}$ and $\mathcal{V}_{\mathrm{t}}$ represent the vocabularies of the input and output sentences, respectively. $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$ denote the one-hot vectors of the i-th and j-th tokens in input and output sentences, respectively, i.e. $\boldsymbol{x}_i \in \{0,1\}^{|\mathcal{V}_{\mathrm{s}}|}$ and $\boldsymbol{y}_j \in \{0,1\}^{|\mathcal{V}_{\mathrm{t}}|}$. Here, we introduce a short notation $\boldsymbol{x}_{i:j}$ for representing a sequence of vectors $(\boldsymbol{x}_i, \ldots, \boldsymbol{x}_j)$. To explain the NMT model concisely, we assume that its input and output are both sequences of one-hot vectors $\boldsymbol{x}_{1:I}$ and $\boldsymbol{y}_{1:J}$ that correspond to input and output sentences whose lengths are I and J, respectively. Thus, the NMT model approximates the following conditional probability:

$$p(\boldsymbol{Y}|\boldsymbol{X}) = \prod_{j=1}^{J+1} p(\boldsymbol{y}_j|\boldsymbol{y}_{0:j-1}, \boldsymbol{X}), \quad (1)$$

where $\boldsymbol{y}_0$ and $\boldsymbol{y}_{J+1}$ represent one-hot vectors of special beginning-of-sentence (BOS) and end-of-sentence (EOS) tokens, respectively, and $\boldsymbol{X} = \boldsymbol{x}_{1:I}$ and $\boldsymbol{Y} = \boldsymbol{y}_{1:J+1}$.

Let $\boldsymbol{E} \in \mathbb{R}^{D\times|\mathcal{V}_{\mathrm{s}}|}$ and $\boldsymbol{F} \in \mathbb{R}^{D\times|\mathcal{V}_{\mathrm{t}}|}$ be the encoder and decoder embedding matrices, respectively, where D is the dimension of the embedding vectors. Thus, $p(\boldsymbol{y}_j|\boldsymbol{y}_{0:j-1}, \boldsymbol{X})$ in Eq. 1 is calculated as follows:

$$\begin{aligned} p(\boldsymbol{y}_j|\boldsymbol{y}_{0:j-1}, \boldsymbol{X}) &= \texttt{AttDec}\big(\boldsymbol{f}_j, \boldsymbol{h}_{1:I}\big), \\ \boldsymbol{h}_{1:I} &= \texttt{Enc}(\boldsymbol{e}_{1:I}), \\ \boldsymbol{f}_j = \boldsymbol{F}\boldsymbol{y}_{j-1}, &\quad \boldsymbol{e}_i = \boldsymbol{E}\boldsymbol{x}_i, \end{aligned} \quad (2)$$

where $\texttt{Enc}(\cdot)$ and $\texttt{AttDec}(\cdot)$ represent functions that abstract the entire encoder and decoder (with an attention mechanism) procedures, respectively.

Training Phase Let $\mathcal{D}$ be the training data consisting of a set of pairs of $\boldsymbol{X}_n$ and $\boldsymbol{Y}_n$, namely, $\mathcal{D} = \{(\boldsymbol{X}_n, \boldsymbol{Y}_n)\}_{n=1}^{N}$, where N represents the amount of training data. For training, we generally seek the optimal parameters $\hat{\boldsymbol{\Theta}}$ that can minimize the following optimization problem:

$$\hat{\boldsymbol{\Theta}} = \operatorname*{argmin}_{\boldsymbol{\Theta}} \big\{\mathcal{J}(\mathcal{D}, \boldsymbol{\Theta})\big\}, \quad (3)$$

$$\mathcal{J}(\mathcal{D}, \boldsymbol{\Theta}) = -\frac{1}{|\mathcal{D}|} \sum_{(\boldsymbol{X},\boldsymbol{Y})\in\mathcal{D}} \ell(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{\Theta}), \quad (4)$$

$$\ell(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{\Theta}) = \log\big(p(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{\Theta})\big), \quad (5)$$

where $\boldsymbol{\Theta}$ represents a set of trainable parameters in the NMT model.

Generation Phase We generally use a K-best beam search to generate an output sentence with the (approximated) K-highest probability given input sentence $\boldsymbol{X}$ in the generation (test) phase. We omit to explain this part in detail as our focus is a regularization technique that is independent of the generation phase.

4 Adversarial Regularization

This section briefly describes the adversarial regularization technique applied to the text classification tasks proposed in Miyato et al. (2017). Let $\hat{\boldsymbol{r}}_i \in \mathbb{R}^D$ be an adversarial perturbation vector for the i-th word in input $\boldsymbol{X}$. The perturbed input embedding $\boldsymbol{e}'_i \in \mathbb{R}^D$ is computed for each encoder time-step i as follows:

$$\boldsymbol{e}'_i = \boldsymbol{E}\boldsymbol{x}_i + \hat{\boldsymbol{r}}_i. \tag{6}$$

4.1 Adversarial Training (AdvT)

To obtain the worst case perturbations as an adversarial perturbation in terms of minimizing the log-likelihood of given $\boldsymbol{X}$, we seek the optimal solution $\hat{\boldsymbol{r}}$ by maximizing the following equation:

$$\hat{\boldsymbol{r}} = \underset{\boldsymbol{r}, ||\boldsymbol{r}|| \leq \epsilon}{\operatorname{argmax}} \Big\{ \ell(\boldsymbol{X}, \boldsymbol{r}, \boldsymbol{Y}, \boldsymbol{\Theta}) \Big\}, \tag{7}$$

where ϵ is a scalar hyper-parameter that controls the norm of the perturbation, and $\boldsymbol{r}$ represents a concatenated vector of $\boldsymbol{r}_i$ for all i. Here, $\ell(\boldsymbol{X}, \boldsymbol{r}, \boldsymbol{Y}, \boldsymbol{\Theta})$ represents an extension of Eq. 5, where the perturbation $\boldsymbol{r}_i$ in $\boldsymbol{r}$ is applied to the position of $\hat{\boldsymbol{r}}_i$ as described in Eq. 6.

However, it is generally infeasible to exactly estimate $\hat{\boldsymbol{r}}$ in Eq. 7 for deep neural models. As a solution, an approximation method was proposed by Goodfellow et al. (2015), where $\ell(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{r}, \boldsymbol{\Theta})$ is linearized around $\boldsymbol{X}$. This approximation method induces the following non-iterative solution for calculating $\hat{\boldsymbol{r}}_i$ for all encoder time-step i:

$$\hat{\boldsymbol{r}}_i = \epsilon \frac{\boldsymbol{a}_i}{||\boldsymbol{a}||_2}, \quad \boldsymbol{a}_i = \nabla_{\boldsymbol{e}_i} \ell(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{\Theta}). \tag{8}$$

Thus, based on adversarial perturbation $\hat{\boldsymbol{r}}$, the loss function can be defined as:

$$\mathcal{A}(\mathcal{D}, \boldsymbol{\Theta}) = -\frac{1}{|\mathcal{D}|} \sum_{(\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{D}} \ell(\boldsymbol{X}, \hat{\boldsymbol{r}}, \boldsymbol{Y}, \boldsymbol{\Theta}). \tag{9}$$

Finally, we jointly minimize the objective functions $\mathcal{J}(\mathcal{D}, \boldsymbol{\Theta})$ and $\mathcal{A}(\mathcal{D}, \boldsymbol{\Theta})$:

$$\hat{\boldsymbol{\Theta}} = \underset{\boldsymbol{\Theta}}{\operatorname{argmin}} \Big\{ \mathcal{J}(\mathcal{D}, \boldsymbol{\Theta}) + \lambda \mathcal{A}(\mathcal{D}, \boldsymbol{\Theta}) \Big\}, \tag{10}$$

where λ is a scalar hyper-parameter that controls the balance of the two loss functions.

4.2 Virtual Adversarial Training (VAT)

Miyato et al. (2016) proposed *virtual adversarial training*, which is mainly used for the semi-supervised extension of the adversarial regularization technique. The difference appears in the loss function ℓ in Eq. 7 and 9. Specifically, we can use perturbations calculated based on the virtual adversarial training by substituting ℓ with the following loss function:

$$\ell_{\mathtt{KL}}(\boldsymbol{X}, \hat{\boldsymbol{r}}, \cdot, \boldsymbol{\Theta}) = \mathtt{KL}\big(p(\cdot \mid \boldsymbol{X}, \boldsymbol{\Theta}) || p(\cdot \mid \boldsymbol{X}, \hat{\boldsymbol{r}}, \boldsymbol{\Theta})\big), \tag{11}$$

where $\mathtt{KL}(\cdot||\cdot)$ denotes the KL divergence.

It is worth noting here that, in our experiments, we never applied the semi-supervised learning, but used the above equation for calculating perturbation as the replacement of standard adversarial regularization. This means that the training data is identical in both settings.

5 Adversarial Regularization in NMT

As strictly following the original definition of the conventional adversarial training, the straightforward approach to applying the adversarial perturbation is to add the perturbation into the encoder-side embeddings $\boldsymbol{e}_i$ as described in Eq. 6. However, NMT models generally have another embedding layer in the decoder-side, as we explained in Eq. 2. This fact immediately offers us also to consider applying the adversarial perturbation into the decoder-side embeddings $\boldsymbol{f}_j$.

For example, let $\hat{\boldsymbol{r}}'_j \in \mathbb{R}^D$ be an adversarial perturbation vector for the j-th word in output $\boldsymbol{Y}$. The perturbed embedding $\boldsymbol{f}'_j \in \mathbb{R}^D$ is computed for each decoder time-step j as follows:

$$\boldsymbol{f}'_j = \boldsymbol{F}\boldsymbol{y}_{j-1} + \hat{\boldsymbol{r}}'_j. \tag{12}$$

Then similar to Eq. 8, we can calculate $\hat{\boldsymbol{r}}'$ as:

$$\hat{\boldsymbol{r}}'_j = \epsilon \frac{\boldsymbol{b}_j}{||\boldsymbol{b}||_2}, \quad \boldsymbol{b}_j = \nabla_{\boldsymbol{f}_j} \ell(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{\Theta}), \tag{13}$$

where $\boldsymbol{b}$ is a concatenated vector of $\boldsymbol{b}_j$ for all j. In addition, we need to slightly modify the definition of $\boldsymbol{r}$, which is originally the concatenation vector of all $\boldsymbol{r}_i$ for all i, to the concatenation vector of all $\boldsymbol{r}_i$ and $\boldsymbol{r}'_j$ for all i and j.

Finally, we have three options for applying the perturbation into typical NMT models, namely, applying the perturbation into embeddings in the (1) encoder-side only, (2) decoder-side only, and (3) both encoder and decoder sides.

	DE↔EN	FR↔EN
training	189,318	208,323
test2012 (dev)	1,700	1,124
test2013 (test)	993	1,024
test2014 (test)	1,305	1,305

Table 1: Number of sentences in our datasets (Datasets are cleaned from the original dataset).

Model	Perturbation position	EN→DE test2013	EN→DE test2014
LSTM	(None)	27.73	23.98
+AdvT	enc-emb	**28.73**	**24.90**
	dec-emb	27.44	23.71
	enc-dec-emb	28.47	24.78
+VAT	enc-emb	29.03	24.75
	dec-emb	27.49	23.20
	enc-dec-emb	**29.47**	**24.92**
Transformer	(None)	29.15	25.19
+AdvT	enc-emb	29.04	25.16
	dec-emb	28.95	25.75
	enc-dec-emb	**29.61**	**25.78**
+VAT	enc-emb	29.95	26.00
	dec-emb	29.62	25.88
	enc-dec-emb	**30.13**	**26.06**

Table 2: BLEU scores averaged over five models in various configurations of perturbation positions (enc-emb, dec-emb, or enc-dec-emb) and adversarial regularization techniques (AdvT or VAT).

6 Experiments

6.1 Datasets

We conducted experiments on the IWSLT evaluation campaign dataset (Cettolo et al., 2012). We used the IWSLT 2016 training set for training models, 2012 test set (test2012) as the development set, and 2013 and 2014 test sets (test2013 and test2014) as our test sets. Table 1 shows the statistics of datasets used in our experiments.

For preprocessing of our experimental datasets, we used the Moses tokenizer[2] and the truecaser[3]. We removed sentences over 50 words from the training set. We also applied the byte-pair encoding (BPE) based subword splitting script[4] with 16,000 merge operations (Sennrich et al., 2016b).

6.2 Model Configurations

We selected two widely used model architectures, namely, LSTM-based encoder-decoder used in Luong et al. (2015) and self-attention-based encoder-decoder, the so-called Transformer (Vaswani et al., 2017). We adapted the hyper-parameters based on the several recent previous papers[5].

Hereafter, we refer to the model trained with the adversarial regularization (ℓ in Eq. 7) as AdvT, and similarly, with the virtual adversarial training (ℓ_{KL} in Eq. 11) as VAT. We set $\lambda = 1$ and $\epsilon = 1$ for all AdvT and VAT experiments.

6.3 Results

Investigation of effective configuration Table 2 shows the experimental results with configurations of perturbation positions (enc-emb, dec-emb, or enc-dec-emb) and adversarial regularization techniques (AdvT or VAT). As evaluation metrics, we used BLEU scores (Papineni et al., 2002)[6]. Note that all reported BLEU scores are averaged over five models.

Firstly, in terms of the effective perturbation position, enc-dec-emb configurations, which add perturbations to both encoder and decoder embeddings, consistently outperformed other configurations, which used either encoder or decoder only. Moreover, we achieved better performance when we added perturbation to the encoder-side (enc-emb) rather than the decoder-side (dec-emb).

Furthermore, the results of VAT was consistently better than those of AdvT. This tendency was also observed in the results reported by Miyato et al. (2016). As discussed in Kurakin et al. (2017), AdvT generates the adversarial examples from correct examples, and thus, the models trained by AdvT tend to overfit to training data rather than those trained by VAT. They referred to this phenomenon of AdvT as *label leaking*.

Results on four language pairs Table 3 shows the BLEU scores of averaged over five models on four different language pairs (directions), namely German→English, French→English, English→German, and English→French. Furthermore, the row (b) shows the results obtained when we incorporated pseudo-parallel corpora generated using the back-translation method (Sennrich et al., 2016a) as additional training data. For

[2] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/tokenizer.perl

[3] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/recaser/truecase.perl

[4] https://github.com/rsennrich/subword-nmt

[5] The detailed hyper-parameters are listed in Appendix A.

[6] We used the multi-bleu.perl script in the Moses toolkit: https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl

	Model	Perturbation position	DE→EN test2013	DE→EN test2014	FR→EN test2013	FR→EN test2014	EN→DE test2013	EN→DE test2014	EN→FR test2013	EN→FR test2014
(a)	LSTM	(None)	32.71	28.53	39.09	36.25	27.73	23.98	38.89	36.18
	Transformer	(None)	34.22	30.19	38.87	37.20	29.15	25.19	40.43	37.90
	+VAT	enc-dec-emb	35.06	**31.10**	40.09	37.89	**30.13**	26.06	41.13	38.64
	+VAT +AdvT	enc-dec-emb	**35.50**	30.88	**40.26**	**38.44**	30.04	**26.33**	**41.67**	**38.72**
(b) w/ BT	Transformer	enc-dec-emb	35.44	31.08	40.44	38.42	30.73	26.02	41.74	39.03
	+VAT	enc-dec-emb	36.43	**32.53**	41.29	**39.76**	**31.99**	**27.20**	**43.41**	**40.15**
	+VAT +AdvT	enc-dec-emb	**36.49**	32.39	**41.56**	39.64	31.29	27.05	42.61	39.95

Table 3: BLEU scores averaged over five models in four different language pairs (directions). (b) Results with using training data increased by back-translation method (BT).

Input	meine gebildete Mutter aber wurde Lehrerin .
Reference	but my educated mother became a teacher .
Baseline (Transformer)	my educated mother , though , became a teacher .
Proposed (Transformer+VAT w/ BT)	but my educated mother became a teacher .
Input	aber man kann sehen , wie die Menschen miteinander kommunizieren , zu welchen Zeiten sie einander anrufen , wann sie zu Bett gehen .
Reference	but you can see how your people are communicating with each other , what times they call each other , when they go to bed .
Baseline (Transformer)	but you can see how people talk to each other about what time they call each other when they go to bed .
Proposed (Transformer+VAT w/ BT)	but you can see how people communicate with each other , at which time they call each other , when they go to bed .
Input	wer im Saal hat ein Handy dabei ?
Reference	who in the room has a mobile phone with you ?
Baseline (Transformer)	who in the room has a cell phone in it ?
Proposed (Transformer+VAT w/ BT)	who in the room has a cell phone with me ?

Table 4: Example translation from German→English (test2013).

generating the pseudo-parallel corpora, we used the WMT14 news translation corpus.

We observe that Transformer+VAT consistently outperformed the baseline Transformer results in both standard (a) and back-translation (b) settings. We report that VAT did not require us to perform additional heavy hyper-parameter search (excluding the hyper-parameter search in base models). Therefore, we can expect that VAT can improve the translation performance on other datasets and settings with relatively high-confidence.

In addition, the rows +VAT+AdvT show the performance obtained by applying both AdvT and VAT simultaneously. We can further improve the performance in some cases, but the improvement is not consistent among the datasets.

Actual Translation Examples Table 4 shows actual translation examples generated by the models compared in our German→English translation setting. We observe that Transformer+VAT with using training data increased by the back-translation method seems to generate higher quality translations compared with those of the baseline Transformer.

7 Conclusion

This paper discussed the practical usage and benefit of adversarial regularization based on adversarial perturbation in the current NMT models. Our experimental results demonstrated that applying VAT to both encoder and decoder embeddings consistently outperformed other configurations. Additionally, we confirmed that adversarial regularization techniques effectively worked even if we performed them with the training data increased by a back-translation method. We believe that adversarial regularization can be one of the common and fundamental technologies to further improve the translation quality, such as model ensemble, byte-pair encoding, and back-translation.

Acknowledgments

We thank three anonymous reviewers for their helpful comments. We also thank Takeru Miyato, who gave us valuable comments about AdvT/VAT.

References

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and Natural Noise Both Break Neural Machine Translation. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*.

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. WIT3: Web Inventory of Transcribed and Translated Talks. In *Proceedings of the 16th Annual Conference of the European Association for Machine Translation (EAMT)*, pages 261–268.

Sumit Chopra, Michael Auli, and Alexander M. Rush. 2016. Abstractive Sentence Summarization with Attentive Recurrent Neural Networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 93–98.

Kevin Clark, Minh-Thang Luong, Christopher D. Manning, and Quoc Le. 2018. Semi-Supervised Sequence Modeling with Cross-View Training. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1914–1925.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*.

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*.

Jianshu Ji, Qinlong Wang, Kristina Toutanova, Yongen Gong, Steven Truong, and Jianfeng Gao. 2017. A Nested Attention Neural Hybrid Model for Grammatical Error Correction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 753–762.

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 (EMNLP)*.

Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. 2017. Adversarial Machine Learning at Scale. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*.

Minh-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 (EMNLP)*, pages 1412–1421.

Takeru Miyato, Andrew M Dai, and Ian Goodfellow. 2017. Adversarial Training Methods for Semi-Supervised Text Classification. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*.

Takeru Miyato, Shin ichi Maeda, Masanori Koyama, Ken Nakae, and Shin Ishii. 2016. Distributional Smoothing with Virtual Adversarial Training. In *Proceedings of the 4th International Conference on Learning Representations (ICLR)*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–318.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A Neural Attention Model for Abstractive Sentence Summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 379–389.

Suranjana Samanta and Sameep Mehta. 2017. Towards Crafting Text Adversarial Samples. *arXiv preprint arXiv:1707.02812*.

Motoki Sato, Jun Suzuki, Hiroyuki Shindo, and Yuji Matsumoto. 2018. Interpretable Adversarial Perturbation in Input Embedding Space for Text. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4323–4330.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving Neural Machine Translation Models with Monolingual Data. In *Proceedings of 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 86–96.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1715–1725.

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural Responding Machine for Short-Text Conversation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL & IJCNLP)*, pages 1577–1586.

Jun Suzuki, Sho Takase, Hidetaka Kamigaito, Makoto Morishita, and Masaaki Nagata. 2018. An Empirical Study of Building a Strong Baseline for Constituency Parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 612–618.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Proceedings of the 31st Annual Conference on Neural Information Processing Systems (NIPS)*, pages 6000–6010.

Oriol Vinyals, Łukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2015. Grammar as a Foreign Language. In *Proceedings of the 29th Annual Conference on Neural Information Processing Systems (NIPS)*, pages 2773–2781.

Dilin Wang, Chengyue Gong, and Qiang Liu. 2019. Improving Neural Language Modeling via Adversarial Training. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pages 6555–6565.