Robust Neural Text Classification and Entailment via Mixup Regularized Adversarial Training

Jiahao Zhao, Penghui Wei, Wenji Mao*

[†]SKL-MCCS, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China

^{*}School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China

{zhaojiahao2019, weipenghui2016, wenji.mao}@ia.ac.cn

ABSTRACT

Recent studies show that neural models for natural language processing are usually fragile under adversarial attacks (e.g., character-level insertion and word-level synonym substitution), which exposes the lack of robustness. Most defense techniques are tailored to specific semantic level attacks and do not possess the ability to mitigate multi-level attack simultaneously. Adversarial training has been shown the effectiveness of increasing model robustness. However, it often suffers from degradation on normal data, especially when the proportion of adversarial examples increase. To address this, we propose mixup regularized adversarial training (MRAT) against multi-level attack. Our method can utilize multiple adversarial examples to increase model intrinsic robustness without sacrificing the performance on normal data. We evaluate our method on text classification and entailment tasks. Experimental results on different text encoders (BERT, LSTM and CNN) with multi-level attack show that our method outperforms adversarial training consistently.

CCS CONCEPTS

• Computing methodologies → Natural language processing; Adversarial learning.

KEYWORDS

Adversarial robustness; Adversarial training; Mixup

ACM Reference Format:

Jiahao Zhao, Penghui Wei, Wenji Mao. 2021. Robust Neural Text Classification and Entailment via Mixup Regularized Adversarial Training. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21)*, July 11–15, 2021, Virtual Event, Canada. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/3404835.3463122>

1 INTRODUCTION

Recent studies show that neural network is brittle under adversarial attack [4, 10, 14]. By adding malicious perturbation to normal data, the attacker could fool the model into making wrong prediction. These perturbed examples are called *adversarial examples* [6].

*Corresponding author

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGIR '21, July 11–15, 2021, Virtual Event, Canada

© 2021 Association for Computing Machinery.

ACM ISBN 978-1-4503-8037-9/21/07...\$15.00

<https://doi.org/10.1145/3404835.3463122>

Table 1: An illustration that shows an original text and its adversarial example with multi-level attack.

	Text (True label: Positive)	Predicted label
Original	flat out amusing, sometimes endearing and often <u>fabulous</u> , with a solid cast, <u>delicious</u> dialogue and a wide supply of <u>effective</u> sight gags	Positive
Adversarial example	flat out amusing, sometimes endearing and ofetn <u>fabulous</u> , with a solid cast, <u>succulent</u> dialogue and a wide supply of <u>usefulness</u> sight gags	Negative

Thus the existence of adversarial examples exposes the robustness concern of deep learning models.

Recent work on adversarial attacks could be divided into two groups, adversarial attacks and defending against them. Most studies in text field focus on adversarial attack that searches small perturbations to fool model. Based on the semantic granularity of perturbation, adversarial attack methods could be grouped into character-level attack [4, 20], word-level attack [10, 21, 26] or sentence-level attack [9, 22, 29]. Table 1 shows an original example and the corresponding adversarial example under multi-level attack that combines multiple granularities of perturbations (word-level synonym substitutions are underlined and character-level typos are double-underlined).

Ideally, model should be robust to these seemingly innocuous modifications. To increase the robustness against adversarial attacks, various defense techniques have been proposed. Most methods are tailored to specific semantic granularity. For instance, Jones et al. [11] use robust encoder to defense character-level attack. Mozes et al. [17] use frequency-guided word substitution to detect word-level attack. Because of the inherent granularity of text, it remains a research challenge to defense multi-level attack simultaneously.

Many studies have shown the potentiality of adversarial training [6], which can increase model robustness on specific semantic granularity. It utilizes generated adversarial examples as data augmentation and retrains the model with a mixture of normal data and adversarial examples. However, due to the memorization of specific adversarial perturbations, adversarial training significantly hurts model performance on the unperturbed normal data, especially when the proportion of adversarial examples increases [25]. This dramatically limits the capability of adversarial training.

To address the above issue, in this paper, we propose mixup regularized adversarial training (MRAT) for defending multi-level attack. Our method employs mixup [27] on multiple adversarial examples, which add linear interpolation between normal data and multiple adversarial examples. Mathematically, Zhang et al. [28] show that mixup training is approximately a regularized loss minimization. This mixup interpolation restricts the model to behave linearly on adversarial examples and implicitly controls model complexity. To

avoid model's memorizing specific adversarial perturbation, our method minimizes the JS-divergence among the distribution of normal samples and mixup interpolated samples. By this means, model's intrinsic adversarial robustness of our method can be enhanced without sacrificing normal data performance.

We conduct extensive experiments on text classification and textual entailment tasks against multi-level attack.¹ Results on different text encoders (BERT, LSTM and CNN) demonstrate that our method can increase model robustness against multi-level adversarial attack and consistently outperforms adversarial training.

The main contributions of our work are summarized as follows:

- We propose a mixup regularized adversarial training (MRAT) method against multi-level attack on text.
- Our method can increase model's intrinsic robustness without sacrificing model performance on normal data.
- Experiments on text classification and entailment tasks verify that the adversarial robustness of our method under multi-level attack consistently surpasses adversarial training over three state-of-the-art encoders.

2 BACKGROUND

Given a textual input $x \in \mathcal{X}$, the goal of model is to make a prediction $y \in \mathcal{Y}$. We denote the training data and the test data as $\mathcal{D}^{\text{train}}$ and $\mathcal{D}^{\text{test}}$ respectively, where $(x, y) \in \mathcal{D}$.

We introduce the workflow of evaluating adversarial robustness. Following the practice by Jin et al. [10], first, we train a model $\mathcal{F}$ on $\mathcal{D}^{\text{train}}$ and evaluate the model accuracy on $\mathcal{D}^{\text{test}}$ to ensure the model is well trained. To evaluate adversarial robustness, we attack the model $\mathcal{F}$ by using the selected attack method $\mathcal{A}$ on $\mathcal{D}^{\text{test}}$. Specifically, for each correct classified example in $\mathcal{D}^{\text{test}}$, attacker searches perturbations δ and adds δ to this example. The perturbed example that successfully fool the model is called adversarial example $\hat{x}$. If the attacker fails to fool the model for this example, the model could still predict the example accurately. In other words, the model is robust to this example. In the end, we use model accuracy under attack to evaluate the model robustness.

Furthermore, for adversarial training, we generate adversarial examples by attacking the normally trained model $\mathcal{F}$ on $\mathcal{D}^{\text{train}}$. To increase model robustness against this adversarial attack, we then use both the original data $(x, y) \in \mathcal{D}^{\text{train}}$ and the adversarial examples $(\hat{x}, y)$ to train the model again. We denote the trained model as $\hat{\mathcal{F}}$. Same as before, we attack the model $\hat{\mathcal{F}}$ on $\mathcal{D}^{\text{test}}$ and use the model accuracy under attack $\mathcal{A}$ as adversarial robustness.

3 PROPOSED METHOD

We propose Mixup Regularized Adversarial Training (MRAT) for enhancing model robustness against multi-level attack. Our method is shown in Algorithm 1. It consists of two main steps. The first step is to generate adversarial examples (line 1-3). The second step is to retrain the model with mixup interpolated examples (line 4-11).

3.1 Generate adversarial examples

Adversarial examples are crucial for adversarial training. In this first step, we train a victim model and generate adversarial examples by using multiple adversarial attack methods to attack the model.

Algorithm 1 Mixup regularized adversarial training

Require: Training data $\mathcal{D}^{\text{train}}$

- 1: Train model $\mathcal{F}$ on $\mathcal{D}^{\text{train}}$
 - 2: Attack model $\mathcal{F}$ by using attack method $\mathcal{A}$ on $\mathcal{D}^{\text{train}}$
 - 3: Form paired data $(x, \hat{x}, y) \in \hat{\mathcal{D}}^{\text{train}}$ with adversarial examples
 - 4: Initialize the model $\mathcal{F}$ with random parameter θ
 - 5: **for** $(x, \hat{x}, y)$ in $\hat{\mathcal{D}}^{\text{train}}$ **do**
 - 6: Sample mixup weight λ from Beta(α, α)
 - 7: $H(\tilde{x}) = \lambda \cdot H(x) + (1 - \lambda) \cdot H(\hat{x})$
 - 8: $p_{\text{orig}} = \mathcal{F}(x), p_{\text{mix}} = \mathcal{F}(\tilde{x})$
 - 9: $\mathcal{L} = \mathcal{L}(y, p_{\text{orig}}) + \lambda_{\text{mix}} \mathbb{D}_{\text{JS}}(p_{\text{orig}} \parallel p_{\text{mix}})$
 - 10: Optimize $\mathcal{L}$
 - 11: **end for**
-

3.1.1 Train a victim model. Given an input text $x \in \mathcal{X}$, the target of the neural model $\mathcal{F}$ is to give a prediction $y \in \mathcal{Y}$. Note that x could be either one sentence (e.g. in text classification) or two paired sentences (e.g. in textual entailment). The neural model $\mathcal{F}$ could be either CNN, LSTM, or BERT. We train a model by minimizing the standard cross-entropy loss.

After training, we test the model on $\mathcal{D}^{\text{test}}$ to ensure that it achieves a reasonable accuracy score.

3.1.2 Attack the victim model. Using selected attack method $\mathcal{A}$, we attack model $\mathcal{F}$ on $\mathcal{D}^{\text{train}}$. For instance, in text classification task, given a textual input x where x is a sequence of words $w_1, w_2, \dots, w_L$. The attack method $\mathcal{A}$ searches allowed perturbation δ like insert, delete, swap or substitute for every w . An adversarial example we generated could be $\hat{x} = (w_1, \hat{w}_2, \dots, w_L)$, where $\hat{w}_2$ is perturbed word, $\arg \max \mathcal{F}(x) = y$, and $\arg \max \mathcal{F}(\hat{x}) \neq y$. When attack is finished, we save all generated adversarial examples.

The attack method could be any semantic granularity of text. We use multiple attack methods to generate adversarial examples.

3.1.3 Form paired adversarial data. In the end, by attacking $(x, y) \in \mathcal{D}^{\text{train}}$, we use generated adversarial examples to form paired data $(x, \hat{x}, y) \in \hat{\mathcal{D}}^{\text{train}}$.

3.2 Retrain the model regularized by mixup

To further extend the usage of various adversarial examples, we leverage mixup with adversarial training. First, we add interpolation between paired normal examples and adversarial examples. Then we retrain the model, which uses normal examples for classification loss and uses interpolated examples for regularization loss.

3.2.1 Mixup interpolation between paired data. Given a paired data $(x, \hat{x}, y) \in \hat{\mathcal{D}}^{\text{train}}$, we interpolate the hidden representation between normal data x and the corresponding adversarial example $\hat{x}$.

$$H(\tilde{x}) = \lambda \cdot H(x) + (1 - \lambda) \cdot H(\hat{x}), \quad (1)$$

where $\tilde{x}$ is the interpolated example, $H(x)$ is the hidden representation of input x produced by the model, and λ is mixup weight between $[0, 1]$. λ is sampled from Beta(α, α), where $\alpha \in (0, \infty)$. We apply mixup interpolation on the last hidden layer.

3.2.2 Mixup regularization. We use Jensen-Shannon divergence among distributions of original samples and mixup interpolated examples as a regularization term.

$$\mathbb{D}_{\text{JS}}(p_{\text{orig}} \parallel p_{\text{mix}}), \quad (2)$$

where $p_{\text{orig}} = \mathcal{F}(x)$ and $p_{\text{mix}} = \mathcal{F}(\tilde{x})$.

¹The code, trained model, and adversarial examples are available at <https://git.io/J3y84>.

Table 2: Results of BERT-based model trained by using multiple adversarial examples. We use all three types of adversarial examples in the training stages. We report model accuracy in original test set and accuracy under attacks.

Approach	Text Classification: MR				Textual Entailment: SNLI			
	Original	Multi-level	Char-level	Word-level	Original	Multi-level	Char-level	Word-level
Norm	86.5	24.0	15.3	4.4	89.1	2.5	6.6	1.7
NormMix	86.1 (↓ -0.4)	31.4 (↑ +7.4)	21.7 (↑ +6.4)	8.3 (↑ +3.9)	89.4 (↑ +0.3)	3.2 (↑ +0.7)	8.5 (↑ +1.9)	3.1 (↑ +1.4)
AT	81.2 (↓ -5.3)	14.4 (↓ -9.6)	14.6 (↓ -0.7)	4.5 (↑ +0.1)	89.0 (↓ -0.1)	5.1 (↑ +2.6)	16.4 (↑ +9.8)	3.2 (↑ +1.5)
MRAT	86.3 (↓ -0.2)	34.9 (↑ +10.9)	25.8 (↑ +10.5)	7.9 (↑ +3.5)	89.5 (↑ +0.4)	9.9 (↑ +7.4)	23.4 (↑ +16.8)	10.5 (↑ +8.8)
MRAT+	86.9 (↑ +0.4)	35.5 (↑ +11.5)	27.1 (↑ +11.8)	7.4 (↑ +3.0)	88.7 (↓ -0.4)	12.2 (↑ 9.7)	26.5 (↑ +19.9)	12.4 (↑ +10.7)

3.2.3 Retrain model. Toward this end, we use the standard loss for normal example (x, y) and the JS-divergence mentioned above to retrain the model.

$$\mathcal{L}(y, p_{\text{orig}}) + \lambda_{\text{mix}} \mathbb{D}_{\text{JS}}(p_{\text{orig}} \parallel p_{\text{mix}}), \quad (3)$$

where $\mathcal{L}(\cdot, \cdot)$ is the cross-entropy loss, $\mathbb{D}_{\text{JS}}$ is the Jensen-Shannon divergence, and λ_{mix} is a hyper-parameter that balances the loss.

4 EXPERIMENT

4.1 Experimental Setup

4.1.1 Evaluation datasets. We conduct experiments on two fundamental NLP tasks: text classification and textual entailment.

- MR: Sentence-level sentiment classification dataset which contains movie reviews collected from Rotten Tomatoes [18]. Each sentence is labeled as positive sentiment or negative sentiment. We follow the practice of Jin et al. [10] which uses 90% data as training set and the left 10% as test set.
- SNLI: Textual entailment dataset [1]. Given a sentence pair (premise, hypothesis), the task is to indicate whether the relationship of the pair is entailment, contradiction, or neutral.

4.1.2 Victim Models. We use BERT [2] as an attacking target model. Each input sentence is tokenized with an additional special token [CLS] at the start of the sequence and [SEP] at the end. We use the representation of token [CLS] to classify the input sentence.

For each dataset, we fine-tune the BERT model on the training set and achieve a reasonable accuracy score on the test set.

Furthermore, we also test WordCNN [12] and LSTM [8] encoder on text classification task. For WordCNN model, we use 3 window sizes (3,4,5) and 100 filters for each window. For LSTM model, we use a 1-layer bidirectional LSTM with hidden state size of 150. Both models use 200-dimensional GloVe [19] embeddings.

4.1.3 Attack methods. We take evaluation on two perturbation levels: character-level and word-level. Specifically, we use three attack methods [16] in the experiment.

- Multi-level attack: textbugger [13] is a multi-level attack method. It combines character and word level perturbations.
- Character-level attack: deepwordbug [5] is a character-level method that uses character swap, insertion, deletion, etc.
- Word-level attack: textfooler [10] is a word-level attack method that uses synonyms substitution.

For textual entailment task, character-level (deepwordbug) and word-level (textfooler) attacks are only allowed to add perturbations to hypothesis. Multi-level (textbugger) attack is set to a more difficult scenario, which both hypothesis and premise are allowed.

For MR, we use all the training set examples to generate adversarial examples. While for SNLI, we use only half of the training set to generate adversarial examples.

4.1.4 Evaluation metrics. Firstly, we evaluate model accuracy in the original test set $\mathcal{D}^{\text{test}}$. To evaluate model robustness, we use the split in [10], which randomly selects 1K samples from test set. We attack each model using all the attack methods mentioned above.

4.1.5 Approaches for comparison. All approaches are listed below:

- Norm: A model trained by the standard training process.
- NormMix: A model trained with the standard mixup, which mixes two randomly selected normal examples.
- AT: A model trained by using adversarial training, which uses a mixture of normal dataset and adversarial examples.
- MRAT: A model trained by using Mixup Regularized Adversarial Training (MRAT).
- MRAT+: On the top of MRAT, it further applies the standard mixup to normal dataset.

We utilize the selected attack method $\mathcal{A}$ to attack the victim model *Norm* on training set $\mathcal{D}^{\text{train}}$ and save generated adversarial examples $\mathcal{D}^{\text{train}}$. For fair comparison, we utilize the same $\mathcal{D}^{\text{train}}$ for each compared approach. The mixing policy is set to $\alpha = 1$.

4.2 Experimental Results and Analysis

4.2.1 Experimental results. We utilize all adversarial examples generated by three attack methods to train the model. To evaluate model robustness, each model is attacked by all three attack methods, respectively. The results of BERT based model are shown in table 2. Original accuracy and accuracy under attack are reported.

The adversarial robustness of our method outperforms baselines by a large margin without sacrificing model performance on normal data. Results are consistent over all three attack methods, which confirm our method is suitable for multi-level attack.

4.2.2 Effects of single type adversarial examples. We also evaluate model performance when only one type of adversarial example is used. For instance, we only use adversarial examples generated by word-level attack in the training stage and only evaluate model robustness against word-level attack. Results are shown in Table 3.

Our method outperforms baselines even when the model only uses single type adversarial examples. But model robustness is lower than using multiple type adversarial examples in Table 2. This indicates that our method could leverage multi-level adversarial examples efficiently to further increase model robustness.

Table 3: Results of BERT-based model. Different from Table 2, we only use one type of attack method to generate adversarial examples. We report model original accuracy and accuracy under that attack (Here Ori. denotes original).

Approach	Text Classification: MR						Textual Entailment: SNLI					
	Ori.	Multi-level	Ori.	Char-level	Ori.	Word-level	Ori.	Multi-level	Ori.	Char-level	Ori.	Word-level
Norm	86.5	23.3	86.5	15.3	86.5	4.4	89.1	2.5	89.1	6.6	89.1	1.7
AT	87.9	13.7	85.3	10.2	81.9	4.9	90.0	6.5	88.8	6.8	89.1	8.6
MRAT	88.3	22.3	87.6	12.3	87.8	5.5	89.2	10.4	88.7	19.7	89.6	15.1
MRAT+	87.6	29.9	87.9	18.1	85.6	7.6	90.9	10.9	90.2	19.4	89.3	13.2

Table 4: Results of different encoders (Ori. denotes original).

Approach	WordCNN		LSTM	
	Ori.	Multi-level Attack	Ori.	Multi-level Attack
Norm	79.1	16.4	81.4	19.0
AT	75.2	14.2 (↓ -2.2)	79.7	16.0 (↓ -3.0)
MRAT	79.2	21.1 (↑ 4.7)	80.4	25.2 (↑ 6.2)
MRAT+	78.9	22.4 (↑ +6.0)	82.2	23.8 (↑ +4.8)

4.2.3 Effects of different encoders. We evaluate performance using WordCNN and LSTM models on text classification task. Table 4 shows the result in MR dataset.

Compared with the results of BERT based model in Table 2, the overall robustness improvements on WordCNN and LSTM models are lower. Even so, our method outperforms baselines consistently, which shows our method is encoder-agnostic.

4.2.4 Effects to attack methods. We report additional metrics for attack methods, which shows how the model robustness affects attack methods. The success rate shows the attack effectiveness. The average number of queries shows the attack efficiency. The percent of perturbed words shows the quality of adversarial examples. The latter two metrics are calculated on successful attacks.

Table 5 shows the attack evaluation metric on MR dataset of BERT-based model. Our method makes the attack effectiveness and efficiency decrease marginally. And the attack needs to perturb more words to break our method which leads to attack quality decrease. Overall, our method is more robust that attacker needs more effort to break our model.

5 RELATED WORKS

Adversarial training. Adversarial training methods could be divided into gradient-based and argumentation-based. The former applies the adversarial gradient to the word embedding layer [3, 15]. Because adversarial attack changes textual input, not word embedding, gradient-based method could merely increase model robustness against adversarial attack. The latter uses the crafted adversarial examples as data augmentation [4, 10, 14, 26]. They only use one single type of adversarial example. We further leverage multiple types of adversarial examples against multi-level attack.

Mixup. Originally from image domain, mixup [27] adds linear interpolation between random examples. This restricts model to behave linearly on out-of-distribution data and implicitly controls model complexity. In text domain, Guo [7] adopts mixup as out-of-manifold data augmentation and uses it to increase the model performance on text classification. Sun et al. [24] apply mixup on

Table 5: Results of attack methods on metrics.

Multi-level: textbugger			
Approach	Success %	# Avg Queries	perturbed word %
Norm	72.25	53.15	15.80
AT	82.27	45.42	14.14
MRAT	59.56	57.60	18.00
MRAT+	59.15	60.67	18.35
Character-level: deepwordbug			
Approach	Success %	# Avg Queries	perturbed word %
Norm	82.31	37.58	18.39
AT	82.02	35.10	14.62
MRAT	70.10	42.05	20.90
MRAT+	68.81	42.99	20.29
Word-level: textfooler			
Approach	Success %	# Avg Queries	perturbed word %
Norm	94.91	100.70	16.16
AT	94.46	106.60	17.92
MRAT	90.85	118.39	19.98
MRAT+	91.48	122.65	20.75

GLUE and increase model generalization empirically. A contemporaneous study [23] also adopts mixup to improve model robustness for text classification. Unlike this work, our motivation is to enhance model robustness against multi-level attack, and thus we apply mixup between paired normal example and corresponding adversarial example other than random operation. Besides, our method is evaluated on both classification and entailment tasks, with different encoders including BERT, CNN and LSTM.

6 CONCLUSION

We propose an effective method MART for improving adversarial robustness of neural NLP models against multi-level attack. Our method leverages mixup as regularization for adversarial training. Compared to adversarial training, our method can increase model robustness without sacrificing model performance on normal data. Experiment results on text classification and textual entailment consistently demonstrate the superiority of our method. Our future work shall explore the mixup representation in deriving tighter certified robustness bounds.

ACKNOWLEDGMENTS

This work was supported in part by the National Key R&D Program of China under Grant #2020AAA0103405, NSFC Grants #11832001 and #71621002, and CAS Strategic Priority Research Program under Grant #XDA27030100.

REFERENCES

- [1] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, 632–642. <https://doi.org/10.18653/v1/D15-1075>
- [2] 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)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. <https://doi.org/10.18653/v1/N19-1423>
- [3] Xin Dong, Yaxin Zhu, Yupeng Zhang, Zuohui Fu, Dongkuan Xu, Sen Yang, and Gerard de Melo. 2020. Leveraging Adversarial Training in Self-Learning for Cross-Lingual Text Classification. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (Virtual Event, China) (SIGIR '20)*. Association for Computing Machinery, New York, NY, USA, 1541–1544. <https://doi.org/10.1145/3397271.3401209>
- [4] Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejeng 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)*. Association for Computational Linguistics, Melbourne, Australia, 31–36. <https://doi.org/10.18653/v1/P18-2006>
- [5] Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-Box Generation of Adversarial Text Sequences to Evade Deep Learning Classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)*. 50–56. <https://doi.org/10.1109/SPW.2018.00016>
- [6] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In *International Conference on Learning Representations*. <http://arxiv.org/abs/1412.6572>
- [7] Hongyu Guo. 2020. Nonlinear Mixup: Out-Of-Manifold Data Augmentation for Text Classification. *Proceedings of the AAAI Conference on Artificial Intelligence* 34, 04 (Apr. 2020), 4044–4051. <https://doi.org/10.1609/aaai.v34i04.5822>
- [8] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (1997), 1735–1780.
- [9] 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)*. Association for Computational Linguistics, New Orleans, Louisiana, 1875–1885. <https://doi.org/10.18653/v1/N18-1170>
- [10] Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is BERT Really Robust? A Strong Baseline for Natural Language Attack on Text Classification and Entailment. *Proceedings of the AAAI Conference on Artificial Intelligence* 34, 05 (Apr. 2020), 8018–8025. <https://doi.org/10.1609/aaai.v34i05.6311>
- [11] Erik Jones, Robin Jia, Aditi Raghunathan, and Percy Liang. 2020. Robust Encodings: A Framework for Combating Adversarial Typos. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 2752–2765. <https://doi.org/10.18653/v1/2020.acl-main.245>
- [12] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, 1746–1751. <https://doi.org/10.3115/v1/D14-1181>
- [13] Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2019. TextBugger: Generating Adversarial Text Against Real-world Applications. *Proceedings 2019 Network and Distributed System Security Symposium (2019)*. <https://doi.org/10.14722/ndss.2019.23138>
- [14] Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. BERT-ATTACK: Adversarial Attack Against BERT Using BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 6193–6202. <https://doi.org/10.18653/v1/2020.emnlp-main.500>
- [15] Takeru Miyato, Andrew M. Dai, and Ian J. Goodfellow. 2017. Adversarial Training Methods for Semi-Supervised Text Classification. In *International Conference on Learning Representations*.
- [16] John X. Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. TextAttack: A Framework for Adversarial Attacks, Data Augmentation, and Adversarial Training in NLP. [arXiv:2005.05909](https://arxiv.org/abs/2005.05909) [cs.CL]
- [17] Maximilian Mozes, Pontus Stenetorp, Bennett Kleinberg, and Lewis Griffin. 2021. Frequency-Guided Word Substitutions for Detecting Textual Adversarial Examples. , 171–186 pages. <https://www.aclweb.org/anthology/2021.eacl-main.13>
- [18] Bo Pang and Lillian Lee. 2005. Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL '05)*. Association for Computational Linguistics, Ann Arbor, Michigan, 115–124. <https://doi.org/10.3115/1219840.1219855>
- [19] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, 1532–1543. <https://doi.org/10.3115/v1/D14-1162>
- [20] Danish Pruthi, Bhuwan Dhingra, and Zachary C. Lipton. 2019. Combating Adversarial Misspellings with Robust Word Recognition. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 5582–5591. <https://doi.org/10.18653/v1/P19-1561>
- [21] Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating Natural Language Adversarial Examples through Probability Weighted Word Saliency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 1085–1097. <https://doi.org/10.18653/v1/P19-1103>
- [22] 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)*. Association for Computational Linguistics, Melbourne, Australia, 856–865. <https://doi.org/10.18653/v1/P18-1079>
- [23] Chenglei Si, Zhengyan Zhang, Fanchao Qi, Zhiyuan Liu, Yasheng Wang, Qun Liu, and Maosong Sun. 2020. Better Robustness by More Coverage: Adversarial Training with Mixup Augmentation for Robust Fine-tuning. [arXiv:2012.15699](https://arxiv.org/abs/2012.15699) [cs.CL]
- [24] Lichao Sun, Congying Xia, Wenpeng Yin, Tingting Liang, Philip Yu, and Lifang He. 2020. Mixup-Transformer: Dynamic Data Augmentation for NLP Tasks. In *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, Barcelona, Spain (Online), 3436–3440. <https://doi.org/10.18653/v1/2020.coling-main.305>
- [25] Wenqi Wang, Lina Wang, Run Wang, Zhibo Wang, and Aoshuang Ye. 2020. Towards a Robust Deep Neural Network in Texts: A Survey. [arXiv:1902.07285](https://arxiv.org/abs/1902.07285) [cs.CL]
- [26] Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun. 2020. Word-level Textual Adversarial Attacking as Combinatorial Optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 6066–6080. <https://doi.org/10.18653/v1/2020.acl-main.540>
- [27] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup: Beyond Empirical Risk Minimization. In *International Conference on Learning Representations*. <https://openreview.net/forum?id=r1Ddp1-Rb>
- [28] Linjun Zhang, Zhun Deng, Kenji Kawaguchi, Amirata Ghorbani, and James Zou. 2021. How Does Mixup Help With Robustness and Generalization?. In *International Conference on Learning Representations*. <https://openreview.net/forum?id=8yKEo06dKNo>
- [29] Zhengli Zhao, Dheeru Dua, and Sameer Singh. 2018. Generating Natural Adversarial Examples. In *International Conference on Learning Representations*. <https://openreview.net/forum?id=H1BLjgZCb>