Better Robustness by More Coverage: Adversarial and Mixup Data Augmentation for Robust Finetuning

Chenglei Si[1,3*], **Zhengyan Zhang**[2,3*], **Fanchao Qi**[2,3], **Zhiyuan Liu**[2,3,4†], **Yasheng Wang**[5], **Qun Liu**[5], **Maosong Sun**[2,3,4]
[1]University of Maryland, College Park, MD, USA
[2]Department of Computer Science and Technology, Tsinghua University, Beijing, China
[3]Beijing National Research Center for Information Science and Technology
[4]Institute for Artificial Intelligence, Tsinghua University, Beijing, China
[5]Huawei Noah's Ark Lab
clsi@terpmail.umd.edu, zy-z19@mails.tsinghua.edu.cn

Abstract

Pretrained language models (PLMs) perform poorly under adversarial attacks. To improve the adversarial robustness, adversarial data augmentation (ADA) has been widely adopted to cover more search space of adversarial attacks by adding textual adversarial examples during training. However, the number of adversarial examples for text augmentation is still extremely insufficient due to the exponentially large attack search space. In this work, we propose a simple and effective method to cover a much larger proportion of the attack search space, called Adversarial and Mixup Data Augmentation (AMDA). Specifically, AMDA linearly interpolates the representations of pairs of training samples to form new virtual samples, which are more abundant and diverse than the discrete text adversarial examples in conventional ADA. Moreover, to fairly evaluate the robustness of different models, we adopt a challenging evaluation setup, which generates a new set of adversarial examples targeting each model. In text classification experiments of BERT and RoBERTa, AMDA achieves significant robustness gains under two strong adversarial attacks and alleviates the performance degradation of ADA on the clean data. Our code is available at: https://github.com/thunlp/MixADA.

1 Introduction

Pretrained language models (PLMs) have established state-of-the-art results on various NLP tasks (Devlin et al., 2019; Liu et al., 2019; Lan et al., 2020) and the pretraining-then-finetuning paradigm has become the status quo. However, recent works have shown the adversarial vulnerabilities of PLMs, where PLMs finetuned on various downstream datasets are fooled by different types of adversarial attacks (Jin et al., 2020; Zang et al., 2020; Si et al., 2021; Li et al., 2020; Garg and Ramakrishnan, 2020; Wang et al., 2020a).

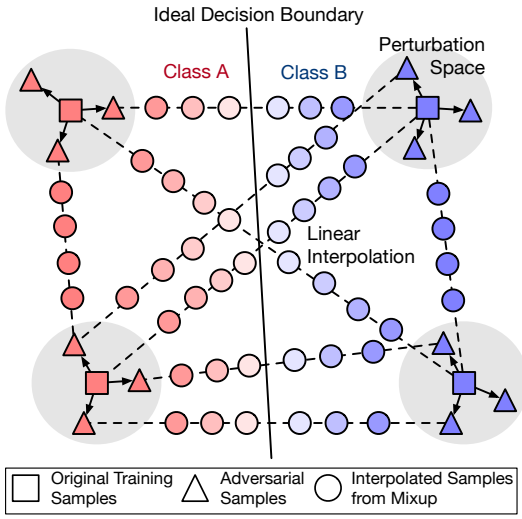

Figure 1: Illustration of MixADA. Some of the interpolated samples are shown. We interpolate the representations of each pair of training samples including original samples and adversarial samples. Blue and red represent two different classes. The solid line represents the resultant decision boundary. AMDA helps achieve a more robust decision boundary.

To improve adversarial robustness, two types of defense strategies have been proposed. The first type targets at specific attacks, such as spelling correction modules and pretraining tasks to defend character-level attacks (Pruthi et al., 2019; Jones et al., 2020; Ma et al., 2020) and certified robustness for word-substitution attacks (Huang et al., 2019; Jia et al., 2019). However, they are limited in practice as they are not generally applicable to other types of attacks. The other type of defense is Adversarial Data Augmentation (ADA), which augments the training set by the adversarial examples and is widely used in the training (finetuning) process to enhance model robustness (Alzantot et al.; Ren et al., 2019; Zhang et al., 2020; Jin et al., 2020;

[*] Equal contribution
[†] Corresponding author email: liuzy@tsinghua.edu.cn

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 1569–1576
August 1–6, 2021.

Li et al., 2020; Tan et al., 2020; Yin et al., 2020; Zheng et al., 2020; Zou et al., 2020; Wang et al., 2020b). ADA is generally applicable to any type of adversarial attacks but is not very effective in improving model performance under attacks. In this work, we aim to improve ADA and devise a general defense strategy to effectively improve model robustness during finetuning.[1]

ADA has two major limitations for NLP models. Firstly, unlike images, it is harder to create new augmented textual data due to their discrete nature. Moreover, for textual adversarial attacks, the attack search space is prohibitively large. For example, the search space of word-substitution attacks consists of all combinations of the synonym replacement candidates, which is exponentially large. Consequently the number of adversarial training examples for augmentation is very insufficient. Secondly, ADA usually causes significant performance degradation on the clean data because the distribution of adversarial examples is very different from that of the clean data (Ren et al., 2019).

In order to solve these two limitations, we create additional training samples via interpolating existing samples (Figure 1). How to interpolate discrete textual inputs is non-trivial. We propose to convert the discrete textual inputs into continuous representations and then perform both ADA and mixup augmentation (Zhang et al., 2018; Guo et al., 2019), which is an augmentation technique proven to be particularly effective on continuous image data (Lamb et al., 2019; Pang et al., 2020). We name our method Adversarial and Mixup Data Augmentation (AMDA). With AMDA, we can create a much larger number of augmented training samples that cannot be obtained via discrete perturbations on textual data. Moreover, AMDA's interpolated virtual training samples are closer to the distribution of the original data, which alleviates the performance degradation problem of ADA.

We experiment AMDA on three text classification datasets under two strong adversarial attacks and find that AMDA achieves significant robustness gains in all cases, notably restoring RoBERTa after-attack accuracy from 6.35% to 51.84% on IMDB, outperforming all other baselines by large margins. Moreover, we also examine the evaluation method for adversarial robustness. Specifically, we find that the widely adopted *Static Attack Evaluation* where a fixed set of adversarial examples are used to test all models is not reliable. In order to test model robustness under targeted attacks (i.e., not model-agnostic), we adopt the more challenging *Targeted Attack Evaluation* where we generate a new set of targeted adversarial examples to evaluate each model. We encourage future defense works to also adopt this more reliable and challenging evaluation setting.

[1] In this paper, we refer to such discrete adversarial training method as adversarial data augmentation to avoid confusion with the gradient-based adversarial training methods (Miyato et al., 2017), which has been shown to be ineffective in defending against textual adversarial attacks (Li and Qiu, 2021).

2 Method

In AMDA, we first augment training samples with ADA and then perform mixup during model training, where mixup augmentation is applied on the ADA-augmented training set.

2.1 Adversarial Data Augmentation

Given a victim model f_v and the original training instances $\mathbf{D}_{ori} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{n}$, we employ an attacker to construct label-preserving adversarial training instances $\mathbf{D}_{adv} = \{(\mathbf{x}'_i, \mathbf{y}_i)\}_{i=1}^{n}$ such that: instances originally correctly classified are now classified wrongly ($f_v(\mathbf{x}'_i) \neq f_v(\mathbf{x}_i)$). We then train the model on the augmented training data $\mathbf{D}_{ADA} = \mathbf{D}_{ori} \cup \mathbf{D}_{adv}$.

2.2 Mixup Data Augmentation

To better defend against the large number of possible adversarial examples, we propose to perform additional mixup augmentation during training. Specifically, we linearly interpolate the representations and labels of pairs of training samples to create different virtual training samples, which can be formulated as:

$$\begin{aligned} \hat{\mathbf{x}} &= \lambda \mathbf{x}_i + (1-\lambda)\mathbf{x}_j, \\ \hat{\mathbf{y}} &= \lambda \mathbf{y}_i + (1-\lambda)\mathbf{y}_j, \end{aligned} \tag{1}$$

where $(\mathbf{x}_i, \mathbf{y}_i)$ and $(\mathbf{x}_j, \mathbf{y}_j)$ are two labeled examples, and $\lambda \in [0, 1]$ comes from a beta distribution $\lambda \sim Beta(\alpha, \alpha)$, where α is a hyperparameter. On textual data, we cannot directly mix the discrete tokens. Instead, we can either interpolate the word embedding vectors or models' hidden representations of textual inputs. Meanwhile, we directly interpolate the labels, which are represented as one-hot vectors.

When applied together with adversarial data augmentation, we allow the mixing of different types

of data (between original examples, between original examples and adversarial examples, and between adversarial examples) to increase diversity.

2.3 AMDA

In our proposed Adversarial and Mixup Data Augmentation (AMDA), we train the new model f on the augmented training data $\mathbf{D}_{AMDA}$, which is obtained by performing both adversarial data augmentation and mixup data augmentation. We minimize the sum of the standard training loss and the mixup loss:

$$L = \sum_{i=1}^{n} L_{CE}(f(\mathbf{x}_i), \mathbf{y}_i) + \sum_{i=1}^{m} L_{KL}(f(\hat{\mathbf{x}}_i), \hat{\mathbf{y}}_i), \quad (2)$$

where $(\mathbf{x}_i, \mathbf{y}_i)$ is from $\mathbf{D}_{ADA}$ and $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ is the virtual example obtained by applying mixup on the random pair of training data sampled from $\mathbf{D}_{ADA}$. We use cross-entropy to compute loss on $(\mathbf{x}_i, \mathbf{y}_i)$ and use KL-divergence for loss on $(\hat{\mathbf{x}}_i, \hat{\mathbf{y}}_i)$.

3 Robustness Evaluation

There are two different ways of robustness evaluation under adversarial attacks used in previous works. In this work, we explicitly differentiate them as Static Attack Evaluation (SAE) and Targeted Attack Evaluation (TAE):

SAE generates a fixed set of adversarial examples on the original model as the victim model. This fixed adversarial test set will then be used to evaluate all the new models. This evaluation setup has been adopted in (Ren et al., 2019; Tan et al., 2020; Yin et al., 2020; Wang et al., 2020b; Zou et al., 2020; Wang et al., 2021, *inter alia.*).

TAE re-generates a new set of adversarial examples to target every model being evaluated. This is adopted in (Zhang et al., 2020; Huang et al., 2019; Jia et al., 2019; Li et al., 2020; Zang et al., 2020; Zheng et al., 2020; Li and Qiu, 2021, *inter alia.*)

We observe that some authors did not explicitly specify the mode of evaluation in their papers[2], leading to confusion and even conflicting conclusions. Thus, we explicitly differentiate the two modes of evaluation and provide a comparison in our experiments.

[2]We had to email some of the authors to clarify the evaluation setup being adopted.

4 Experiments

4.1 Experiment Setups

Datasets. We evaluate our methods on three text classification datasets: two sentiment analysis datasets: SST-2 (Socher et al., 2013) and IMDB (Maas et al., 2011), where both datasets are binary classification tasks; as well as a multi-class news classification dataset AGNews (Zhang et al., 2015), which consists of four different classes. For SST-2, we attack the entire test set (1821 samples) and report the accuracy under attacks. For IMDB, we find that it is prohibitively slow to attack the whole test set (25k samples) and hence we use the subset of the original test set as released in Gardner et al. for faster evaluation, which consists of 488 test instances. Similarly, on AGNews, we randomly sampled 10% of the original test set and hold out as the test samples for attack evaluation. We also include these data splits in our released code base for easy reproduction and fair comparison for future works.

Victim models and attack methods. We experiment with both BERT-base-uncased (Devlin et al., 2019) and RoBERTa-base (Liu et al., 2019) as the victim models. We use PWWS (Ren et al., 2019) and TextFooler (Jin et al., 2020) as our attack methods, which have been shown to effectively attack state-of-the-art NLP models including PLMs such as BERT. Both attack algorithms have access to model predictions but not gradients, and iteratively search for word synonym substitutes that flip model predictions without drastically changing the original semantic meanings and golden labels.

Details of mixup. When performing mixup, we mix hidden representations of upper layers of BERT. The vectors used for mixup are hidden representations of the input examples at layer i of the Transformer encoder, where i is randomly sampled from $\{7, 9, 12\}$, which was found to be empirically effective (Chen et al., 2020). Furthermore, we explore two different ways of obtaining the hidden representations of input examples from PLMs like BERT: (1) We use the vector of the `[CLS]` token at the ith-layer of BERT as the hidden representation for mixing. We name this approach **SMix**. (2) We perform mixup on every token's vector representation at the ith-layer. We name this approach **TMix**, which is the approach taken by Chen et al. (2020).

Details of ADA and AMDA. For both ADA and AMDA, we generate and add the corresponding adversarial examples of PWWS and TextFooler into

	SST-2					IMDB				
	Original	PWWS-d	PWWS-s	TF-d	TF-s	Original	PWWS-d	PWWS-s	TF-d	TF-s
BERT_v	92.04	19.17	19.17	5.66	5.66	97.34	23.36	23.36	3.48	3.48
BERT_{r1}	91.10	18.73	44.98	3.46	45.36	96.72	25.61	69.88	1.64	76.64
BERT_{r2}	90.94	20.26	45.63	2.97	45.80	97.13	30.12	65.78	2.46	76.23
RoBERTa_v	94.45	25.48	25.48	3.29	3.29	97.75	15.98	15.98	1.84	1.84
RoBERTa_{r1}	94.29	31.03	50.47	5.82	41.63	97.54	27.46	65.57	3.48	77.46
RoBERTa_{r2}	93.85	32.13	50.69	9.34	40.91	97.34	17.21	73.36	2.87	76.64

Table 1: Comparison between dynamic and static evaluation. PWWS-d, PWWS-s, TF-d, TF-s represent PWWS dynamic, PWWS static, TextFooler dynamic, TextFooler static, respectively. Numbers in the table represent accuracy. BERT_v and RoBERTa_v are the victim model for generating static evaluation examples. BERT_{r1}, BERT_{r2}, RoBERTa_{r1}, and RoBERTa_{r2} are the fine-tuned models with new random seeds.

	SST-2				IMDB			
	PWWS		TextFooler		PWWS		TextFooler	
	Original	Adversarial	Original	Adversarial	Original	Adversarial	Original	Adversarial
BERT	91.27	14.83 *(20.88%)*	91.27	2.97 *(16.21%)*	97.75	24.18 *(24.10%)*	97.75	1.64 *(10.18%)*
+ADA	90.12	27.18 *(24.46%)*	90.50	9.01 *(18.32%)*	96.93	25.82 *(34.53%)*	96.93	3.07 *(11.81%)*
+TMix	91.82	21.20 *(19.36%)*	91.82	3.51 *(16.39%)*	97.13	43.24 *(32.51%)*	97.13	0.00 *(12.06%)*
+SMix	91.82	22.52 *(20.47%)*	91.82	4.61 *(16.76%)*	97.13	31.97 *(23.74%)*	97.13	2.66 *(12.39%)*
+AMDA-TMix	91.54	**38.82** *(23.73%)*	91.93	13.23 *(19.66%)*	97.34	51.02 *(36.76%)*	96.72	4.51 *(17.23%)*
+AMDA-SMix	91.10	31.52 *(24.11%)*	92.15	**17.35** *(18.64%)*	96.72	**60.86** *(27.79%)*	96.72	**17.42** *(13.85%)*
RoBERTa	94.62	28.39 *(23.06%)*	94.62	5.44 *(18.51%)*	97.54	28.07 *(37.48%)*	97.54	6.35 *(12.61%)*
+ADA	94.07	25.26 *(27.07%)*	92.75	9.67 *(19.71%)*	97.54	24.80 *(49.36%)*	96.93	12.50 *(14.39%)*
+TMix	94.18	30.04 *(23.19%)*	94.18	11.04 *(17.69%)*	97.54	44.06 *(39.33%)*	97.54	21.11 *(14.01%)*
+SMix	93.96	31.52 *(22.86%)*	93.96	8.29 *(17.80%)*	97.34	41.39 *(34.90%)*	97.34	22.34 *(11.96%)*
+AMDA-TMix	93.90	36.74 *(26.02%)*	93.03	13.78 *(20.15%)*	98.57	50.41 *(59.68%)*	97.13	**51.84** *(16.62%)*
+AMDA-SMix	93.96	**41.85** *(27.17%)*	93.47	**16.80** *(21.88%)*	97.54	**55.12** *(45.30%)*	97.54	49.18 *(15.52%)*

Table 2: Accuracy of the various models under PWWS and TextFooler attacks. Best performance for BERT-based models and RoBERTa-based models under each attack is **boldfaced**, the second best performance is underlined. *Numbers* in brackets indicate the average word modification rate of each attack.

	PWWS		TextFooler	
	Orig.	Adv.	Orig.	Adv.
RoBERTa	94.34	47.50	94.34	25.53
+ADA	93.55	66.97	94.08	44.61
+TMix	94.08	45.66	94.08	26.58
+SMix	94.08	45.13	94.08	22.63
+AMDA-TMix	94.47	69.74	93.95	**56.32**
+AMDA-SMix	94.34	**70.00**	93.42	51.32

Table 3: Results on AG News multi-class classification dataset, with RoBERTa model. Best performance under each attack is **boldfaced**, the second best performance is underlined.

training. For comparison, we also experiment with mixup alone without adding the adversarial examples. In this case, the model would only interpolate pairs of original training examples. We perform a greedy hyper-parameter search for the amount of augmented adversarial training samples and mixup parameter α as described in the Appendix. We also report average word modification rates, which indicate the percentage of words being replaced for attacking. Higher word modification rates indicate that the model is harder to attack and hence needs more words to be replaced.

4.2 Comparison of SAE and TAE

To compare SAE and TAE, we attack the fine-tuned model (BERT_v), RoBERTa_v as the victim on SST-2 and IMDB, and then use the generated adversarial test set as the fixed test set for SAE. We then change the random seeds and re-finetune the models on the same data (BERT_{r1}, BERT_{r2}, RoBERTa_{r1}, RoBERTa_{r2}) with all other hyper-parameters being the same. We evaluate all these models using both SAE and TAE. The results are shown in Table 1.

We find that by simply changing the random seeds, models achieve significant improvement under SAE. However, when we re-generate the adversarial test set for each model, their performances under TAE stay consistently poor. Moreover, we train BERT and RoBERTa with ADA and find that although BERT_{ADA} and $\text{RoBERTa}_{\text{ADA}}$ perform

well under SAE, they still perform poorly under TAE. This shows that conventional ADA is actually ineffective in improving model robustness under the challenging TAE setting. We conclude that the adversarial examples found by the attackers target specifically at the victim models, hence they cannot fully reveal weaknesses of new models even if they only differ in random seeds. We believe that TAE is the more challenging and meaningful evaluation method to measure model robustness under targeted attacks. We adopt TAE for the rest of the experiments in this paper and encourage future works to do so for fair comparison.

4.3 Mixup Improves Robustness

The comparison of AMDA and baseline methods under attacks for SST-2 and IMDB is shown in Table 2. The results on the AGNews dataset with RoBERTa model is shown in Table 3. We observe that: (1) Mixup alone (both TMix and SMix) can often improve model robustness. For example, TMix and SMix improve the robust accuracy significantly under both attacks when using RoBERTa on IMDB. (2) AMDA (both AMDA-TMix and AMDA-SMix) can achieve further robustness improvement as compared to ADA and mixup in all cases. This proves that mixup and ADA can complement each other to better improve model robustness under adversarial attacks. (3) Compared to ADA, our AMDA method does not incur significant performance degradation on the original test sets while improving robustness. In some cases, for example, BERT+TMix and BERT+AMDA-TMix even improve the model performance on the original test sets. This benefit is likely because that mixup creates virtual examples that are closer to the empirical data distribution. (4) We find that models trained with AMDA also incur higher word modification rates under both attacks. For example, RoBERTa+AMDA-TMix incurs 59.68% word modification rate under PWWS attack, while the RoBERTa baseline only needs 37.48% words to be replaced. This further demonstrates that our proposed method improves robustness.

5 Conclusion

In this work, we propose AMDA as a generally applicable defense strategy by combining both adversarial and mixup data augmentation to cover more of the attack space. We show that AMDA greatly improves PLMs' robustness under the challenging TAE evaluation setting under two strong adversarial attacks. We leave a more thorough theoretical analysis of AMDA's effectiveness on textual data as future work.[3] We believe that our work can establish the appropriate evaluation protocol and offer a competitive baseline for future works on improving the robustness of PLMs.

Acknowledgments

We thank members of THUNLP for their helpful discussion and valuable feedback on our work. This work is supported by the National Key Research and Development Program of China (No. 2020AAA0106501) and Beijing Academy of Artificial Intelligence (BAAI).

References

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. Generating natural language adversarial examples. In *Proceedings of EMNLP 2018*.

Jiaao Chen, Zichao Yang, and Diyi Yang. 2020. MixText: Linguistically-informed interpolation of hidden space for semi-supervised text classification. In *Proceedings of ACL 2020*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL 2019*.

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. Evaluating models' local decision boundaries via contrast sets. In *Proceedings of EMNLP 2020 (Findings)*.

Siddhant Garg and Goutham Ramakrishnan. 2020. BAE: BERT-based adversarial examples for text classification. In *Proceedings of EMNLP 2020*.

Hongyu Guo, Yongyi Mao, and Richong Zhang. 2019. Augmenting Data with Mixup for Sentence Classification: An Empirical Study. *arXiv*.

Po-Sen Huang, Robert Stanforth, Johannes Welbl, Chris Dyer, Dani Yogatama, Sven Gowal, Krishnamurthy Dvijotham, and Pushmeet Kohli. 2019.

[3] Zhang et al. (2021) provided a theoretical proof of mixup's effectiveness on continuous image data, which may serve as a foundation for more theoretical work.

Achieving verified robustness to symbol substitutions via interval bound propagation. In *Proceedings of EMNLP 2019*.

Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. 2019. Certified Robustness to Adversarial Word Substitutions. In *Proceedings of EMNLP 2019*.

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. In *Proceedings of AAAI 2020*.

Erik Jones, Robin Jia, Aditi Raghunathan, and Percy Liang. 2020. Robust Encodings: A Framework for Combating Adversarial Typos. In *Proceedings of ACL 2020*.

Alex Lamb, Vikas Verma, Juho Kannala, and Yoshua Bengio. 2019. Interpolated Adversarial Training: Achieving Robust Neural Networks Without Sacrificing Too Much Accuracy. In *Proceedings of AISec@CCS*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *Proceedings of ICLR 2020*.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. BERT-ATTACK: Adversarial attack against BERT using BERT. In *Proceedings of EMNLP 2020*.

Linyang Li and Xipeng Qiu. 2021. TAVAT: Token-Aware Virtual Adversarial Training for Language Understanding. In *Proceedings of AAAI 2021*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining approach. *arXiv*.

Wentao Ma, Yiming Cui, Chenglei Si, Ting Liu, Shijin Wang, and Guoping Hu. 2020. CharBERT: Character-aware pre-trained language model. In *Proceedings of COLING 2020*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of ACL 2011*.

Takeru Miyato, Andrew M. Dai, and Ian J. Goodfellow. 2017. Adversarial Training Methods for Semi-Supervised Text Classification. In *Proceedings of ICLR 2017*.

Tianyu Pang, Kun Xu, and Jun Zhu. 2020. Mixup inference: Better exploiting mixup to defend adversarial attacks. In *Proceedings of ICLR 2020*.

Danish Pruthi, Bhuwan Dhingra, and Zachary C. Lipton. 2019. Combating Adversarial Misspellings with Robust Word Recognition. In *Proceedings of ACL 2019*.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating Natural Language Adversarial Examples through Probability Weighted Word Saliency. In *Proceedings of ACL 2019*.

Chenglei Si, Ziqing Yang, Yiming Cui, Wentao Ma, Ting Liu, and Shijin Wang. 2021. Benchmarking Robustness of Machine Reading Comprehension Models. In *Proceedings of ACL 2021 (Findings)*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of EMNLP 2013*.

Samson Tan, Shafiq Joty, Min-Yen Kan, and Richard Socher. 2020. It's morphin' time! Combating linguistic discrimination with inflectional perturbations. In *Proceedings of ACL 2020*.

Boxin Wang, Hengzhi Pei, Boyuan Pan, Qian Chen, Shuohang Wang, and Bo Li. 2020a. T3: Tree-autoencoder constrained adversarial text generation for targeted attack. In *Proceedings of EMNLP 2020*.

Boxin Wang, Shuohang Wang, Y. Cheng, Zhe Gan, Ruoxi Jia, Bo Li, and Jingjing Liu. 2021. InfoBERT: Improving Robustness of Language Models from An Information Theoretic Perspective. In *Proceedings of ICLR 2021*.

Tianlu Wang, Xuezhi Wang, Yao Qin, Ben Packer, Kang Li, Jilin Chen, Alex Beutel, and Ed Chi. 2020b. CAT-gen: Improving robustness in NLP models via controlled adversarial text generation. In *Proceedings of EMNLP 2020*.

Fan Yin, Quanyu Long, Tao Meng, and Kai-Wei Chang. 2020. On the Robustness of Language Encoders against Grammatical Errors. In *Proceedings of ACL 2020*.

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 ACL 2020*.

Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup: Beyond empirical risk minimization. In *Processings of ICLR 2018*.

Huangzhao Zhang, Hao Zhou, Ning Miao, and Lei Li. 2020. Generating Fluent Adversarial Examples for Natural Languages. In *Proceedings of ACL 2019*.

Linjun Zhang, Zhun Deng, Kenji Kawaguchi, Amirata Ghorbani, and James Zou. 2021. How Does Mixup Help With Robustness and Generalization? In *Proceedings of ICLR 2021*.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level Convolutional Networks for Text Classification. In *Proceedings of NeurIPS 2015*.

Xiaoqing Zheng, Jiehang Zeng, Yi Zhou, Cho-Jui Hsieh, Minhao Cheng, and Xuanjing Huang. 2020. Evaluating and enhancing the robustness of neural network-based dependency parsing models with adversarial examples. In *Proceedings of ACL 2020*.

Wei Zou, Shujian Huang, John Xie, Xin-Yu Dai, and Jiajun Chen. 2020. A Reinforced Generation of Adversarial Samples for Neural Machine Translation. In *Proceedings of ACL 2020*.

Appendix

Hyper-parameter Analysis

In this section, we perform further analysis to examine the effects of different hyper-parameters. There are two hyper-parameters involved in MixADA: the amount of adversarial data added for training, and the α parameter in the beta distribution of mixup coefficient. We also experiment with an alternative ADA strategy - iterative ADA.

Amount of Adversarial Training Data

We vary the ratio of the training dataset that we generate adversarial training samples on and add to the MixADA fine-tuning. We experiment with SMixADA with the hyper-parameter of mixup being fixed. On SST-2, we vary the ratio in $\{25\%, 50\%, 75\%, 100\%\}$. On IMDB, since the average sequence length is significantly longer and the adversarial example generation process becomes much slower, we experiment with a set of smaller ratios: $\{5\%, 10\%, 15\%, 20\%\}$. The results are plotted in in Figure 2. Interestingly, we find that higher ratio of adversarial training samples does not necessarily bring in additional robustness gains.

Interpolation Coefficient in Mixup

We also analyse the hyper-parameter of mixup: the α parameter in the beta distribution, from which the interpolation coefficient is sampled. We fix the ratio of adversarial training data and vary α in the range of $\{0.2, 0.4, 2.0, 4.0, 8.0\}$. The results are plotted in Figure 3. We find that there is no consistent pattern across different datasets on what is the optimal α. Hence, for our main experiments in the paper, we perform a greedy hyper-parameter search: we first tune the ratio of adversarial training samples, then fix the ratio and tune the α parameter for mixup. A more exhaustive hyper-parameter search might bring additional performance gains but would also incur extra computation costs.

Iterative ADA

For our MixADA experiments in the paper, we generate all adversarial training samples at one shot and mix them with the original examples before fine-tuning. An alternative is to generate a new batch of adversarial training samples dynamically with the current model at each epoch. We compare this iterative approach with our MixADA and use the same ratio of adversarial training samples and

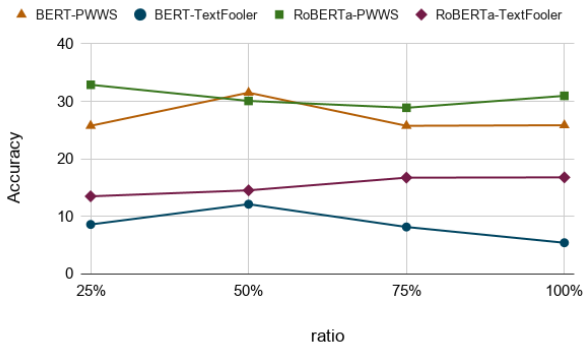

Figure 2: Performance under attacks on the SST-2 dataset with varying ratio of adversarial training samples.

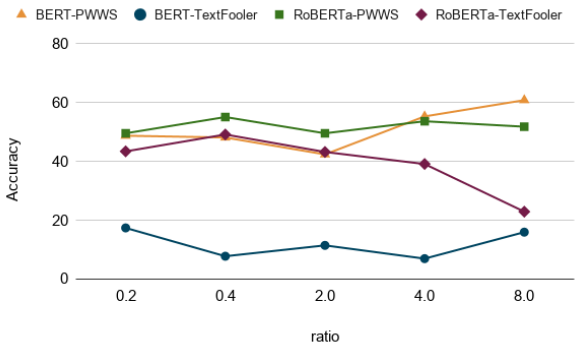

Figure 3: Performance under attacks on the IMDB dataset with varying α parameter for mixup.

mixup parameter α. We evaluate RoBERTa on the SST-2 dataset. The results are in Table 4.

	PWWS	TextFooler
TMixADA	36.74	13.78
+iterative	28.45	6.26
SMixADA	41.85	16.80
+iterative	28.78	7.69

Table 4: Performance of MixADA under attacks in the one-shot approach and the iterative approach.

We find that the iterative approach is far worse than our one-shot approach. We hypothesize that in the one-shot approach, we generate the adversarial examples on a fully-fine-tuned model while the iterative approach generates adversarial examples on the not-well-fine-tuned model in the first few epochs, and hence the adversarial examples generated in the iterative approach are not as challenging and useful as those in our one-shot approach.