In and Out-of-Domain Text Adversarial Robustness via Label Smoothing

Yahan Yang[*]
University of Pennsylvania
yangy96@seas.upenn.edu

Soham Dan[*]
IBM Research
soham.dan@ibm.com

Dan Roth
University of Pennsylvania
danroth@seas.upenn.edu

Insup Lee
University of Pennsylvania
lee@cis.upenn.edu

Abstract

Recently it has been shown that state-of-the-art NLP models are vulnerable to adversarial attacks, where the predictions of a model can be drastically altered by slight modifications to the input (such as synonym substitutions). While several defense techniques have been proposed, and adapted, to the discrete nature of text adversarial attacks, the benefits of general-purpose regularization methods such as label smoothing for language models, have not been studied. In this paper, we study the adversarial robustness provided by label smoothing strategies in foundational models for diverse NLP tasks in both in-domain and out-of-domain settings. Our experiments show that label smoothing significantly improves adversarial robustness in pre-trained models like BERT, against various popular attacks. We also analyze the relationship between prediction confidence and robustness, showing that label smoothing reduces over-confident errors on adversarial examples.

1 Introduction

Neural networks are vulnerable to adversarial attacks: small perturbations to the input ,which do not fool humans (Szegedy et al., 2013; Goodfellow et al., 2014; Madry et al., 2017). In NLP tasks, previous studies (Alzantot et al., 2018; Jin et al., 2019; Li et al., 2020; Garg and Ramakrishnan, 2020) demonstrate that simple word-level text attacks (synonym substitution, word insertion/deletion) easily fool state-of-the-art models, including pre-trained transformers like BERT (Devlin et al., 2019; Wolf et al., 2020). Further, it has recently been shown models are overconfident[1] on examples which are easy to attack (Qin et al., 2021) and indeed, such over-confident predictions plague much of modern deep learning (Kong et al., 2020; Guo et al., 2017; Nguyen et al., 2015; Rahimi et al., 2020). Label smoothing is a regularization method that has been proven effective in a variety of applications, and modalities (Szegedy et al., 2016; Chorowski and Jaitly, 2017; Vaswani et al., 2017). Importantly, it has been shown to reduce overconfident predictions and produce better confidence calibrated classifiers (Muller et al., 2019; Zhang et al., 2021; Dan and Roth, 2021; Desai and Durrett, 2020; Huang et al., 2021; Liu and JaJa, 2020).

In this work, we focus on the question: *does label smoothing also implicitly help in adversarial robustness?* While there has been some investigation in this direction for adversarial attacks in computer vision, (Fu et al., 2020; Goibert and Dohmatob, 2019; Shafahi et al., 2019), there is a gap in understanding of whether it helps with discrete, text adversarial attacks used against NLP systems. With the increasing need for robust NLP models in safety-critical applications and a lack of generic robustness strategies,[2] there is a need to understand inherent robustness properties of popular label smoothing strategies, and the interplay between confidence and robustness of a model.

In this paper, we extensively study standard label smoothing and its adversarial variant, covering robustness, prediction confidence, and domain transfer properties. We observe that label smoothing provides implicit robustness against adversarial examples. Particularly, we focus on pre-trained transformer models and test robustness under various kinds of black-box and white-box word-level adversarial attacks, in both in-domain and out-of-domain scenarios. Our experiments show that label smoothing (1) improves robustness to text adversarial attacks (both black-box and white-box), and (2) mitigates over-confident errors on adversarial textual examples. Analysing the adversarial exam-

[*] The first two authors contributed equally to this paper. Most of the work done while Soham Dan was at the University of Pennsylvania.

[1] Confidence on an example is the highest softmax score of the classifier prediction on that example.

[2] which are flexible, simple and not over-specialized to very specific kinds of text adversarial attacks.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 2: Short Papers, pages 657–669
July 9-14, 2023 ©2023 Association for Computational Linguistics

ples along various quality dimensions reveals the remarkable efficacy of label smoothing as a simple add-on robustness and calibration tool.

2 Background

2.1 Text Adversarial Attacks

Our experiments evaluate the robustness of text classification models under three state-of-the-art text adversarial attacks TextFooler (black-box), BAE (black-box) and SemAttack (white-box), described below.[3]For a particular victim NLP model and a raw text input, the attack produces semantically-similar adversarial text as output. Importantly, only those examples are attacked, which are originally correctly predicted by the victim model. The attacks considered are word-level, i.e. they replace words in a clean text with their synonyms to maintain the meaning of the clean text, but change the prediction of the victim models.

- **TextFooler (TF)**: (Jin et al., 2019) proposes an attack which determines the word importance in a sentence, and then replaces the important words with qualified synonyms.
- **BAE**: (Garg and Ramakrishnan, 2020) uses masked pre-trained language models to generate replacements for the important words until the victim model's prediction is incorrect.
- **SemAttack (SemAtt)**: (Wang et al., 2022) introduces an attack to search perturbations in the contextualized embedding space by formulating an optimization problem as in (Carlini and Wagner, 2016). We specifically use the white-box word-level version of this attack.

2.2 Label Smoothing

Label Smoothing is a modified fine-tuning procedure to address overconfident predictions. It introduces uncertainty to smoothen the posterior distribution over the target labels. Label smoothing has been shown to implicitly calibrate neural networks on out-of-distribution data, where *calibration* measures how well the model confidences are aligned with the empirical likelihoods (Guo et al., 2017).

- **Standard Label Smoothing (LS)** (Szegedy et al., 2013; Muller et al., 2019) constructs a new target vector (y_i^{LS}) from the one-hot target vector (y_i), where $y_i^{LS} = (1-\alpha)y_i + \alpha/K$ for a K class classification problem. α is a hyperparameter selection and its range is from 0 to 1.
- **Adversarial Label Smoothing (ALS)** (Goibert and Dohmatob, 2019) constructs a new target vector (y_i^{ALS}) with a probability of $1-\alpha$ on the target label and α on the label to which the classification model assigns the minimum softmax scores, thus introducing uncertainty.

For both LS and ALS, the cross entropy loss is subsequently minimized between the model predictions and the modified target vectors y_i^{LS}, y_i^{ALS}.

3 Experiments

In this section, we present a thorough empirical evaluation on the effect of label smoothing on adversarial robustness for two pre-trained transformer models: BERT and its distilled variant, dBERT, which are the victim models. [4] We attack the victim models using TF, BAE, and SemAttack. For each attack, we present results on both the standard models and the label-smoothed models on various classification tasks: text classification and natural language inference. For each dataset we evaluate on a randomly sampled subset of the test set (1000 examples), as done in prior work (Li et al., 2021; Jin et al., 2019; Garg and Ramakrishnan, 2020). We evaluate on the following tasks, and other details about the setting is in Appendix A.8:

- **Text Classification**: We evaluate on movie review classification using Movie Review (MR) (Pang and Lee, 2005) and Stanford Sentiment Treebank (SST2) (Socher et al., 2013) (both binary classification), restaurant review classification: Yelp Review (Zhang et al., 2015a) (binary classification), and news category classification: AG News (Zhang et al., 2015b) (having the following four classes: World, Sports, Business, Sci/Tech).
- **Natural Language Inference:** We investigate two datasets for this task: the Stanford Natural Language Inference Corpus (SNLI) (Bowman et al., 2015) and the Multi-Genre Natural Language Inference corpus (MNLI) (Williams et al., 2018), both having three classes. For MNLI, our work only evaluates performance

[3] The black-box attacks keep querying the model with its attempts until the victim model is fooled while the white-box attack has access to the gradients to the model. Further details of the attacks are in (Jin et al., 2019; Garg and Ramakrishnan, 2020; Wang et al., 2022).

[4] Additional results on more datasets, models, other attacks and α values, are presented in the Appendix.

SST-2	Clean Acc (↑)		Attack Success Rate (↓)		Adv Conf (↓)	
BERT(α)	0	0.45	0	0.45	0	0.45
TF	91.97	**92.09**	96.38	**88.92**	78.43	**63.62**
BAE	91.97	**92.09**	57.11	**53.42**	86.92	**68.35**
SemAtt	91.97	**92.09**	86.41	**54.05**	80.12	**64.55**
dBERT(α)	0	0.45	0	0.45	0	0.45
TF	89.56	**89.68**	96.29	**89.77**	76.28	**61.60**
BAE	89.56	**89.68**	59.28	**56.52**	83.55	**66.11**
SemAtt	89.56	**89.68**	91.68	**69.69**	78.93	**62.42**

AG_news	Clean Acc (↑)		Attack Success Rate (↓)		Adv Conf (↓)	
BERT(α)	0	0.45	0	0.45	0	0.45
TF	**94.83**	94.67	88.26	**77.47**	59.02	**42.46**
BAE	**94.83**	94.67	74.83	**62.82**	60.66	**43.98**
SemAtt	**94.83**	94.67	52.65	**30.49**	62.32	**44.99**
dBERT(α)	0	0.45	0	0.45	0	0.45
TF	**94.73**	94.47	90.11	**74.52**	57.60	**41.40**
BAE	**94.73**	94.47	77.79	**63.65**	60.01	**42.74**
SemAtt	**94.73**	94.47	52.07	**34.05**	60.40	**43.27**

Yelp	Clean Acc (↑)		Attack Success Rate (↓)		Adv Conf (↓)	
BERT(α)	0	0.45	0	0.45	0	0.45
TF	**97.73**	97.7	99.32	**92.90**	64.85	**55.36**
BAE	**97.73**	97.7	55.35	**45.14**	68.28	**57.38**
SemAtt	**97.73**	97.7	93.55	**36.17**	74.53	**60.24**
dBERT(α)	0	0.45	0	0.45	0	0.45
TF	**97.47**	97.4	99.45	**93.36**	61.75	**54.63**
BAE	**97.47**	97.4	58.14	**45.59**	64.27	**57.14**
SemAtt	**97.47**	97.4	97.37	**43.92**	71.34	**60.57**

SNLI	Clean Acc (↑)		Attack Success Rate (↓)		Adv Conf (↓)	
BERT(α)	0	0.45	0	0.45	0	0.45
TF	**89.56**	89.23	96.5	**96.15**	68.27	**52.61**
BAE	**89.56**	89.23	74.95	**74.82**	76.13	**57.42**
SemAtt	**89.56**	89.23	99.11	**91.94**	75.41	**58.01**
dBERT(α)	0	0.45	0	0.45	0	0.45
TF	**87.27**	87.1	98.12	**96.86**	65.19	**50.80**
BAE	**87.27**	87.1	74.08	**72.91**	72.89	**55.49**
SemAtt	**87.27**	87.1	98.43	**92.84**	71.17	**54.96**

Table 1: Comparison of standard models and models fine-tuned with standard label smoothing techniques (LS) against various attacks for in-domain data. We show clean accuracy, attack success rate and average confidence on successful adversarial texts. For each dataset, the left column are the results for standard model, and the right column are for LS models where α denotes the label smoothing factor (α=0: no LS). ↑ (↓) denotes higher (lower) is better respectively. dBERT denotes the distilBERT model.

on the matched genre test-set in the OOD setting presented in subsection 3.2 .

3.1 In-domain Setting

In the in-domain setting (iD), the pre-trained transformer models are fine-tuned on the train-set for each task and evaluated on the corresponding test-set. For each case, we report the clean accuracy, the adversarial attack success rate (percentage of misclassified examples after an attack) and the average confidence on successfully attacked examples (on which the model makes a wrong prediction).[5] Table 1 shows the performance of BERT and dBERT, with and without label-smoothing. We choose label smoothing factor $\alpha = 0.45$ for standard label-smoothed models in our experiments.

We see that label-smoothed models are more robust for every adversarial attack across different datasets in terms of the attack success rate, which is a standard metric in this area (Li et al., 2021; Lee et al., 2022). Additionally, the higher confidence of the standard models on the successfully attacked examples indicates that label smoothing helps mitigate overconfident mistakes in the adversarial setting. Importantly, the clean accuracy remains almost unchanged in all the cases. Moreover, we observe that the models gain much more robustness from LS under white-box attack, compared to the black-box setting. We perform hyperparameter sweeping for the label smoothing factor α to investigate their impact to model accuracy and adversarial robustness. Figure 1 shows that the attack success rate gets lower as we increase the label smooth factor when fine-tuning the model while the test accuracy is comparable[6]. However, when the label smoothing factor is larger than 0.45, there is no further improvement on adversarial robustness in terms of attack success rate. Automatic search for an optimal label smoothing factor and its theoretical analysis is important future work.

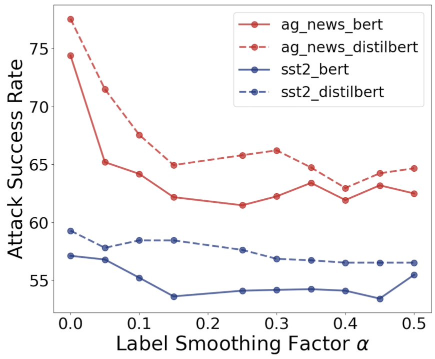

Figure 1: Adversarial success rate versus label smoothing factors (on AG News and SST-2 with BAE attack.)

We also investigate the impact of adversarial label smoothing (ALS) and show that the adversarial label smoothed methods also improves model's robustness in Table 2.

[5] Details of each metric are presented in Appendix A.2.

[6] More results for different α values are in Appendix A.9

SNLI	Clean Acc (↑)		Attack Success Rate (↓)		Adv Conf (↓)	
BERT(α)	0	0.45	0	0.45	0	0.45
TF	**89.56**	88.5	96.5	96.5	68.27	**41.22**
BAE	**89.56**	88.5	74.95	**74.87**	76.13	**44.93**
SemAtt	**89.56**	88.5	99.11	**91.53**	75.41	**44.97**
AG_news	Clean Acc (↑)		Attack Success Rate (↓)		Adv Conf (↓)	
BERT(α)	0	0.45	0	0.45	0	0.45
TF	**94.83**	94.37	88.26	**77.74**	59.02	**32.87**
BAE	**94.83**	94.37	74.83	**64.15**	60.66	**33.45**
SemAtt	**94.83**	94.37	52.65	**27.13**	62.32	**34.72**

Table 2: Comparison of standard models versus models trained with ALS against various attacks on SNLI and AG_news. ↑ (↓) denotes higher (lower) is better respectively.

3.2 Out-of-Domain setting

We now evaluate the benefits of label smoothing for robustness in the out-of-domain (OOD) setting, where the pre-trained model is fine-tuned on a particular dataset and is then evaluated directly on a different dataset, which has a matching label space. Three examples of these that we evaluate on are the Movie Reviews to SST-2 transfer, the SST-2 to Yelp transfer, and the SNLI to MNLI transfer.

In Table 3, we again see that label-smoothing helps produce more robust models in the OOD setting although with less gain compared to iD setting. This is a challenging setting, as evidenced by the significant performance drop in the clean accuracy as compared to the in-domain setting. We also see that the standard models make over-confident errors on successfully attacked adversarial examples, when compared to label-smoothed models.

MR→SST2	Clean Acc (↑)		Attack Success Rate (↓)		Adv Conf (↓)	
BERT (α)	0	0.45	0	0.45	0	0.45
TF	90.71	**91.06**	**90.9**	90.93	69.47	**58.41**
BAE	90.71	**91.06**	**62.83**	63.1	75.2	**62.6**
SemAtt	90.71	**91.06**	82.68	**76.07**	67.64	**57.9**
dBERT(α)	0	0.45	0	0.45	0	0.45
TF	88.19	**88.99**	**94.28**	94.59	64.95	**57.2**
BAE	88.19	**88.99**	**65.41**	65.72	71.89	**61.5**
SemAtt	88.19	**88.99**	88.56	**86.21**	66.51	**58.14**
SNLI→MNLI	Clean Acc (↑)		Attack Success Rate (↓)		Adv Conf (↓)	
BERT (α)	0	0.45	0	0.45	0	0.45
TF	**73.4**	72.1	94.82	**92.79**	58.04	**46.43**
BAE	**73.4**	72.1	82.56	**80.72**	63.00	**49.45**
SemAtt	**73.4**	72.1	99.73	**98.75**	60.32	**47.35**
dBERT(α)	0	0.45	0	0.45	0	0.45
TF	**65.4**	62.1	94.50	**92.59**	54.54	**44.81**
BAE	**65.4**	62.1	77.68	**75.52**	58.88	**47.83**
SemAtt	**65.4**	62.1	99.39	**96.78**	57.10	**45.43**
SST-2 → Yelp	Clean Acc (↑)		Attack Success Rate (↓)		Adv Conf (↓)	
BERT (α)	0	0.45	0	0.45	0	0.45
TF	**92.5**	92.4	99.57	**98.27**	60.80	**54.28**
BAE	**92.5**	92.4	63.68	**60.71**	64.27	**55.66**
SemAtt	**92.5**	92.4	95.80	**68.17**	68.37	**57.45**
dBERT(α)	0	0.45	0	0.45	0	0.45
TF	**91.7**	91.1	99.78	**98.02**	59.12	**53.30**
BAE	**91.7**	91.1	68.70	**63.45**	61.37	**54.21**
SemAtt	**91.7**	91.1	99.02	**82.15**	67.01	**57.37**

Table 3: Comparison of standard models and LS models for various attacks on OOD data where α denotes the label smoothing factor (α=0: no LS).

3.3 Qualitative Results

In this section, we try to understand how the generated adversarial examples differ for label smoothed and standard models. First we look at some qualitative examples: in Table 4, we show some examples (clean text) for which the different attack schemes fails to craft an attack for the label smoothed model but successfully attacks the standard model.

Victim	Attack	Text	
SST2 BERT	BAE	clean adv	at once half-baked and overheated. at once warm and overheated .
MR dBERT	TF	clean adv	no surprises . no surprise .

Table 4: Examples for which an attack could be found for the standard model but not for the label smoothed model. The Victim column shows the dataset and the pretrained model (dBERT denotes distilBERT).

We also performed automatic evaluation of the quality of the adversarial examples for standard and label smoothed models, adopting standard metrics from previous studies (Jin et al., 2019; Li et al., 2021). Ideally, we want the adversarial sentences to be free of grammar errors, fluent, and semantically similar to the clean text. This can be quantified using metrics such as grammar errors, perplexity, and similarity scores (compared to the clean text). The reported scores for each metric are computed over only the successful adversarial examples, for each attack and model type.[7]

SST-2	Perplexity (↑)		Similarity Score (↓)		Grammar Error (↑)	
BERT (α)	0	0.45	0	0.45	0	0.45
TF	400.31	**447.58**	0.800	**0.779**	0.33	**0.38**
BAE	300.74	**305.28**	0.867	**0.855**	−0.05	**−0.04**
AG_News	Perplexity (↑)		Similarity Score (↓)		Grammar Error (↑)	
BERT (α)	0	0.45	0	0.45	0	0.45
TF	342.02	**355.87**	0.782	**0.772**	1.37	**1.40**
BAE	169.37	**170.73**	0.851	**0.845**	0.97	**1.00**

Table 5: Evaluation of adversarial text examples. The results in bold indicates worse adversarial attack quality.

[7] Additional details can be found in AppendixA.3.

Table 5 shows that the quality of generated adversarial examples on label smoothed models is worse than those on standard models for different metrics, suggesting that the adversarial sentences generated by standard models are easier to perceive. This further demonstrates that label smoothing makes it harder to find adversarial vulnerabilities.

4 Conclusion

We presented an extensive empirical study to investigate the effect of label smoothing techniques on adversarial robustness for various NLP tasks, for various victim models and adversarial attacks. Our results demonstrate that label smoothing imparts implicit robustness to models, even under domain shifts. This first work on the effects of LS for text adversarial attacks, complemented with prior work on LS and implicit calibration (Desai and Durrett, 2020; Dan and Roth, 2021), is an important step towards developing robust, reliable models. In the future, it would be interesting to explore the combination of label smoothing with other regularization and adversarial training techniques to further enhance the adversarial robustness of NLP models.

5 Limitations

One limitation of our work is that we focus on robustness of pre-trained transformer language models against word-level adversarial attacks, which is the most common setting in this area. Future work could extend this empirical study to other types of attacks (for example, character-level and sentence-level attacks) and for diverse types of architectures. Further, it will be very interesting to theoretically understand how label smoothing provides (1) the implicit robustness to text adversarial attacks and (2) mitigates over-confident predictions on the adversarially attacked examples.

6 Ethics Statement

Adversarial examples present a severe risk to machine learning systems, especially when deployed in real-world risk sensitive applications. With the ubiquity of textual information in real-world applications, it is extremely important to defend against adversarial examples and also to understand the robustness properties of commonly used techniques like Label Smoothing. From a societal perspective, by studying the effect of this popular regularization strategy, this work empirically shows that it helps robustness against adversarial examples in in-domain and out-of-domain scenarios, for both white-box and black-box attacks across diverse tasks and models. From an ecological perspective, label smoothing does not incur any additional computational cost over standard fine-tuning emphasizing its efficacy as a general-purpose tool to improve calibration and robustness.

Acknowledgements

Research was sponsored by the Army Research Office and was accomplished under Grant Number W911NF-20-1-0080. This work was supported by Contract FA8750-19-2-0201 with the US Defense Advanced Research Projects Agency (DARPA). The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense, the Army Research Office or the U.S. Government. This research was also supported by a gift from AWS AI for research in Trustworthy AI.

References

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. *arXiv preprint arXiv:1804.07998*.

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*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Nicholas Carlini and David Wagner. 2016. Towards evaluating the robustness of neural networks.

Jan Chorowski and Navdeep Jaitly. 2017. Towards better decoding and language model integration in sequence to sequence models. *Proc. Interspeech 2017*, pages 523–527.

Soham Dan and Dan Roth. 2021. On the Effects of Transformer Size on In- and Out-of-Domain Calibration. In *Findings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Shrey Desai and Greg Durrett. 2020. Calibration of pre-trained transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 295–302.

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

Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Chaohao Fu, Hongbin Chen, Na Ruan, and Weijia Jia. 2020. Label smoothing and adversarial robustness. *arXiv preprint arXiv:2009.08233*.

Siddhant Garg and Goutham Ramakrishnan. 2020. Bae: Bert-based adversarial examples for text classification. *arXiv preprint arXiv:2004.01970*.

Morgane Goibert and Elvis Dohmatob. 2019. Adversarial robustness via adversarial label-smoothing.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR.

Chuan Guo, Alexandre Sablayrolles, Hervé Jégou, and Douwe Kiela. 2021. Gradient-based adversarial attacks against text transformers. *arXiv preprint arXiv:2104.13733*.

Shuangping Huang, Yu Luo, Zhenzhou Zhuang, Jin-Gang Yu, Mengchao He, and Yongpan Wang. 2021. Context-aware selective label smoothing for calibrating sequence recognition model. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 4591–4599.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2019. Is bert really robust? natural language attack on text classification and entailment. *arXiv preprint arXiv:1907.11932*.

Lingkai Kong, Haoming Jiang, Yuchen Zhuang, Jie Lyu, Tuo Zhao, and Chao Zhang. 2020. Calibrated language model fine-tuning for in-and out-of-distribution data. *arXiv preprint arXiv:2010.11506*.

Deokjae Lee, Seungyong Moon, Junhyeok Lee, and Hyun Oh Song. 2022. Query-efficient and scalable black-box adversarial attacks on discrete sequential data via bayesian optimization. In *International Conference on Machine Learning*, pages 12478–12497. PMLR.

Dianqi Li, Yizhe Zhang, Hao Peng, Liqun Chen, Chris Brockett, Ming-Ting Sun, and Bill Dolan. 2021. Contextualized perturbation for textual adversarial attack. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5053–5069, Online. Association for Computational Linguistics.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. Bert-attack: Adversarial attack against bert using bert. *arXiv preprint arXiv:2004.09984*.

Chihuang Liu and Joseph JaJa. 2020. Class-similarity based label smoothing for generalized confidence calibration. In *arXiv preprint arXiv: 2006.14028*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

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.

Rafael Muller, Simon Kornblith, and Geoffrey E Hinton. 2019. When does label smoothing help? *Advances in neural information processing systems*, 32.

Anh Nguyen, Jason Yosinski, and Jeff Clune. 2015. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 427–436.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the ACL*.

Yao Qin, Xuezhi Wang, Alex Beutel, and Ed Chi. 2021. Improving calibration through the relationship with adversarial robustness. *Advances in Neural Information Processing Systems*, 34:14358–14369.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Amir Rahimi, Amirreza Shaban, Ching-An Cheng, Richard Hartley, and Byron Boots. 2020. Intra order-preserving functions for calibration of multi-class neural networks. *Advances in Neural Information Processing Systems*, 33:13456–13467.

Emily Reif, Ann Yuan, Martin Wattenberg, Fernanda B Viegas, Andy Coenen, Adam Pearce, and Been Kim. 2019. Visualizing and measuring the geometry of bert. *Advances in Neural Information Processing Systems*, 32.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Ali Shafahi, Amin Ghiasi, Furong Huang, and Tom Goldstein. 2019. Label smoothing and logit squeezing: a replacement for adversarial training? *arXiv preprint arXiv:1910.11585*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Boxin Wang, Chejian Xu, Xiangyu Liu, Yu Cheng, and Bo Li. 2022. SemAttack: Natural textual attacks via different semantic spaces. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 176–205, Seattle, United States. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Chang-Bin Zhang, Peng-Tao Jiang, Qibin Hou, Yunchao Wei, Qi Han, Zhen Li, and Ming-Ming Cheng. 2021. Delving deep into label smoothing. *IEEE Transactions on Image Processing*, 30:5984–5996.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015a. Character-level Convolutional Networks for Text Classification. *arXiv:1509.01626 [cs]*.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015b. Character-level convolutional networks for text classification. In *NIPS*.

A Appendix

- **A.1** Pictorial Overview of the Adversarial Attack Framework
- **A.2** Description of the Evaluation Metrics
- **A.3** Details of Automatic Attack Evaluation
- **A.4** Additional results on Movie Review Dataset
- **A.5** Additional white-box attack on label-smoothed models
- **A.6** Additional results for $\alpha = 0.1$
- **A.7** Additional results on ALBERT model
- **A.8** Dataset overview and expertiment details
- **A.9** Attack success rate versus label smoothing factors for different attacks (TextFooler and SemAttack)
- **A.10** Average number of word change versus Confidence

A.1 Overview of the Framework

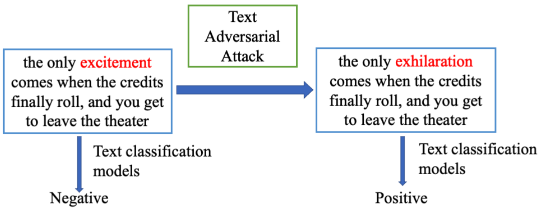
Text Adversarial Attack
the only excitement comes when the credits finally roll, and you get to leave the theater
the only exhilaration comes when the credits finally roll, and you get to leave the theater
Text classification models
Negative
Text classification models
Positive

Figure 2: Here we show an example generated by word-level adversarial attack TextFooler (Jin et al., 2019) on SST-2 data. By replacing excitement with its synonym *exhilaration*, the text classification model changes its prediction from Negative to Positive, which is incorrect.

A.2 Evaluation Metrics

The followings are the details of evaluation metrics from previous works (Lee et al., 2022; Li et al., 2021):

$$\text{Clean accuracy} = \frac{\#\text{ of correctly predicted clean examples}}{\#\text{ of clean examples}}$$

$$\text{Attack Succ. Rate} = \frac{\#\text{ of successful adversarial examples}}{\#\text{ of correctly predicted clean examples}}$$

where successful adversarial examples are derived from correctly predicted examples

$$\text{Adv Conf} = \frac{\text{sum of confidence of successful adv examples}}{\#\text{ of successful adversarial examples}}$$

A.3 Attack evaluation

We performed automatic evaluation of adversarial attacks against standard models and label smoothed models following previous studies (Jin et al., 2019; Li et al., 2021). Following are the details of the metrics we used in Table 5:

Perplexity evaluates the fluency of the input using language models. We use GPT-2 (Radford et al., 2019) to compute perplexity as in (Li et al., 2021) .

Similarity Score determines the similarity between two sentences. We use Sentence Transformers (Reimers and Gurevych, 2019) to compute sentence embeddings and then calculate cosine similarity score between the clean examples and the corresponding adversarially modified examples.

Grammar Error The average grammar error increments between clean examples and the corresponding adversarially modified example.[8]

A.4 Additional results on Movie Review Dataset

Here we provide results of movie review datasets (Pang and Lee, 2005) under in-domain setting.

MR	Clean Acc (↑)		Attack Success Rate (↓)		Adv Conf (↓)	
BERT (α)	0	0.45	0	0.45	0	0.45
TextFooler	**84.4**	83.7	92.54	**92.0**	67.93	**58.33**
BAE	**84.4**	83.7	62.09	**61.17**	74.33	**62.4**
SemAtt	**84.4**	83.7	83.18	**76.34**	68.8	**58.18**
distilBERT(α)	0	0.45	0	0.45	0	0.45
TextFooler	82.3	**82.6**	**94.9**	95.88	64.64	**57.17**
BAE	82.3	**82.6**	67.31	**67.19**	70.54	**60.88**
SemAtt	82.3	**82.6**	90.16	**87.77**	65.55	**57.33**

Table 6: Comparison of standard models and label smoothed models against various attacks for Movie Review dataset.

A.5 Additional results on an additional white-box attack

In this section, we use another recent popular white-box attack named Gradient-based Attack (Guo et al., 2021). This is a gradient-based approach that searches for a parameterized word-level adversarial attack distribution, and then samples adversarial examples from the distribution. We run this attack on standard and label smoothed BERT models and the results are listed below.

We observe that the label smoothing also help with adversarial robustness against this attack across four datasets under iD setting. The results also show that, similar to SemAttack, the grad-based attack benefits more from label smoothing compared to black-box attacks like TextFooler and BAE.

Grad Attack	Clean Acc (↑)		Attack Succ Rate(↓)		Adv Conf (↓)	
BERT (α)	0	0.45	0	0.45	0	0.45
SST-2	91.97	**92.09**	98.38	**82.94**	98.75	**76.35**
AG_news	**94.9**	94.8	98.63	**68.88**	95.35	**63.25**
Yelp	95.3	**95.5**	99.90	**87.02**	99.24	**76.52**
SNLI	89.7	**90.2**	96.1	**86.36**	59.99	**37.28**
SST2 → Yelp	**88.6**	88.4	99.89	**94.84**	98.37	**77.52**

Table 7: Comparison of standard models and label smoothed BERT models against gradient-based attack across different datasets.

A.6 Additional results of $\alpha = 0.1$

Table 8 and 9 are the additional results to show when label smoothing $\alpha = 0.1$, how the adversarial robustness of fine-tuned language models changes under iD and OOD scenarios.

Table 10 are the additional results for adversarial label smoothing $\alpha = 0.1$.

A.7 Additional results on ALBERT

In this section, we include experiment results for standard ALBERT and label smoothed ALBERT in Table 11. We observe that the label smoothing technique also improves adversarial robustness of ALBERT model across different datasets.

SST-2	Clean Acc (↑)		Attack Success Rate (↓)		Adv Conf (↓)	
α	0	0.45	0	0.45	0	0.45
TF	92.66	**92.78**	94.68	**90.73**	76.29	**65.63**
BAE	92.66	**92.78**	**60.15**	65.02	83.67	**70.17**
AG_news	Clean Acc (↑)		Attack Success Rate (↓)		Adv Conf (↓)	
α	0	0.45	0	0.45	0	0.45
TF	**94.9**	94.5	77.66	**56.72**	58.78	**42.59**
BAE	**94.9**	94.5	65.54	**49.74**	59.98	**43.79**
SNLI	Clean Acc (↑)		Attack Success Rate (↓)		Adv Conf (↓)	
α	0	0.45	0	0.45	0	0.45
TF	90.1	**90.3**	94.89	**93.69**	69.66	**53.67**
BAE	90.1	**90.3**	76.91	**75.86**	75.05	**56.42**

Table 11: Comparison of standard models and label smoothed models against TextFooler and BAE attacks for ALBERT model.

[8] we use `https://pypi.org/project/language-tool-python/` to compute grammar error.

SST-2	Clean Acc (↑)		Attack Success Rate (↓)		Adv Conf (↓)	
BERT (α)	0	0.1	0	0.1	0	0.1
TF	91.97	**92.2**	96.38	**94.4**	78.43	**74.39**
BAE	91.97	**92.2**	57.11	**55.22**	86.92	**82.29**
distilBERT(α)	0	0.1	0	0.1	0	0.1
TF	89.56	**89.68**	96.29	**95.14**	76.28	**70.77**
BAE	89.56	**89.68**	59.28	**58.44**	83.55	**78.16**
AG_news	Clean Acc (↑)		Attack Success Rate (↓)		Adv Conf (↓)	
BERT (α)	0	0.1	0	0.1	0	0.1
TF	94.83	**95.0**	88.26	**78.39**	59.02	**55.17**
BAE	94.83	**95.0**	74.83	**65.58**	60.66	**56.24**
distilBERT(α)	0	0.1	0	0.1	0	0.1
TF	**94.73**	94.53	90.11	**81.66**	57.6	**53.43**
BAE	**94.73**	94.53	74.83	**67.7**	60.01	**54.64**
Yelp	Clean Acc (↑)		Attack Success Rate (↓)		Adv Conf (↓)	
BERT (α)	0	0.1	0	0.1	0	0.1
TF	97.73	**97.77**	99.32	**97.99**	64.85	**63.18**
BAE	97.73	**97.77**	55.35	**52.88**	68.28	**66.28**
distilBERT(α)	0	0.1	0	0.1	0	0.1
TF	97.47	**97.5**	99.45	**98.91**	61.75	**60.35**
BAE	97.47	**97.5**	58.14	**51.86**	64.27	**63.04**
SNLI	Clean Acc (↑)		Attack Success Rate (↓)		Adv Conf (↓)	
BERT (α)	0	0.1	0	0.1	0	0.1
TF	**89.56**	88.87	**96.5**	96.74	68.83	**64.96**
BAE	**89.56**	88.87	**74.95**	75.1	76.13	**72.65**
distilBERT(α)	0	0.1	0	0.1	0	0.1
TF	**87.27**	87.03	98.12	**96.94**	65.19	**62.41**
BAE	**87.27**	87.03	74.08	**73.82**	72.89	**69.57**

Table 8: Comparison of standard models and label smoothed models against various attacks for in-domain data where α denotes the label smoothing factor, 0 indicating no LS. [9] ↑ (↓) denotes higher (lower) is better respectively.

SNLI → MNLI	Clean Acc (↑)		Attack Success Rate (↓)		Adv Conf (↓)	
BERT (α)	0	0.1	0	0.1	0	0.1
TextFooler	**73.4**	71.9	**94.82**	94.85	58.04	**48.56**
BAE	**73.4**	71.9	82.56	**77.19**	63	**49.3**
distilBERT(α)	0	0.45	0	0.45	0	0.45
TextFooler	**65.4**	65.2	94.5	**94.17**	54.54	**52.63**
BAE	**65.4**	65.2	77.68	**75.15**	58.88	**56.16**
SST-2 → Yelp	Clean Acc (↑)		Attack Success Rate (↓)		Adv Conf (↓)	
BERT (α)	0	0.1	0	0.1	0	0.1
TextFooler	**92.5**	92.0	99.57	**99.13**	60.8	**58.13**
BAE	**92.5**	92.0	63.68	**63.37**	64.27	**60.63**
distilBERT(α)	0	0.45	0	0.45	0	0.45
TextFooler	**91.7**	91.4	99.78	**99.34**	59.12	**56.42**
BAE	**91.7**	91.4	68.7	**67.07**	61.37	**57.73**

Table 9: Comparison of standard models versus label smoothed models against various attacks for OOD data where α denotes the label smoothing factor (α=0: no LS). ↑ (↓) denotes higher (lower) is better respectively.

SNLI	Clean Acc (↑)		Attack Success Rate (↓)		Adv Conf (↓)	
BERT(α)	0	0.1	0	0.1	0	0.1
TF	89.56	**90.4**	96.5	**95.02**	68.27	**67.54**
BAE	89.56	**90.4**	**74.95**	75.96	76.13	**73.83**
AG_news	Clean Acc (↑)		Attack Success Rate (↓)		Adv Conf (↓)	
BERT(α)	0	0.1	0	0.1	0	0.1
TF	**94.83**	94.6	88.26	**85.27**	59.02	**53.17**
BAE	**94.83**	94.6	74.83	**69.1**	60.66	**54.99**

Table 10: Comparison of standard models versus models trained with ALS against various attacks on SNLI and AG_news. ↑ (↓) denotes higher (lower) is better respectively.

Dataset	No. of classes	Train/Test size	Avg. Length
MR	2	8530/1066	18.64
SST-2	2	6.7e4/872	17.4
Yelp	2	5.6e5/3.8e4	132.74
AG_news	4	1.2e5 /7600	38.68
SNLI	3	5.5e5 /1e4	22.01
MNLI	3	3.9e5/ 9815	28.96

Table 12: Summary of datasets

A.8 Dataset Overview and Experiments Details

We use Huggingface (Wolf et al., 2020) to load the dataset and to fine-tune the pre-trained models. All models are fine-tuned for 3 epochs using AdamW optimizer (Loshchilov and Hutter, 2017) and the learning rate starts from $5e-6$. The training and attacking are run on an NVIDIA Quadro RTX 6000 GPU (24GB). For both BAE and Textfooler attack, we use the implementation in TextAttack (Morris et al., 2020) with the default hyper-parameters (Except for AG_news, we relax the similarity threshld from 0.93 to 0.7 when using BAE attack). The SemAttack is implemented by (Wang et al., 2022) while the generating contextualized embedding space is modified from (Reif et al., 2019). The reported numbers are the average performance over 3 random runs of the experiment for iD setting, and the standard deviation is less than 2%.

A.9 Attack success rate versus label smoothing factors

As mentioned in Section 3.1, we plot the attack success rate of BAE attack versus the label smoothing factors. Here, we plot the results for the TextFooler and SemAttack in Figure 3 and 4, and observe the same tendency as we discussed above.

We also plot the attack success rate of

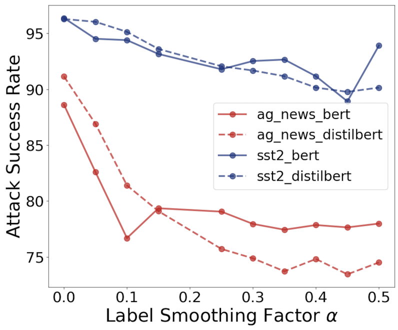

Figure 3: Adversarial success rate versus label smoothing factors for the TextFooler attack (on AG News and SST-2.)

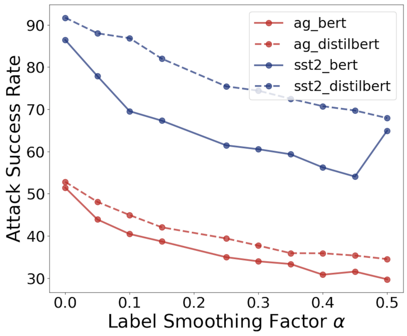

Figure 4: Adversarial success rate versus label smoothing factors for the SemAttack (on AG News and SST-2.)

BAE/TextFooler attack versus the adversarial label smoothing factors in Figure 5 and 6.

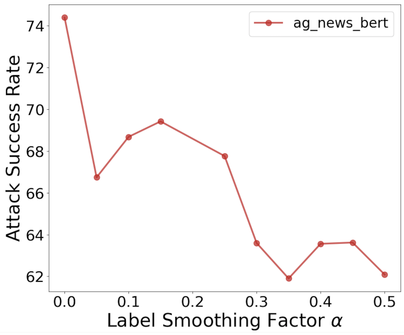

Figure 5: Adversarial success rate versus adversarial label smoothing factors for the BAE attack (on AG News).

We additionally plot the clean accuracy versus the label smoothing factor in Figure 7, and find out that there is not much drop in clean accuracy with increasing the label smoothing factors.

A.10 Average number of word change versus Confidence

Word change rate is defined as the ratio between the number of word replaced after attack and the total number of words in the sentence. Here we plot the bucket-wise word change ratio of adversarial attack versus confidence, and observe that the word change rate for high-confident examples are higher for label smoothed models compared to standard models in most cases. This indicates that it is more difficult to attack label smoothed text classification models. Also note that there is the word change rate is zero because there is no clean texts fall into those two bins.

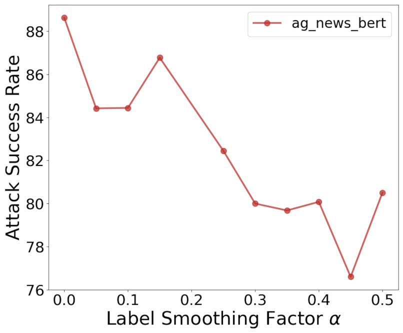

Figure 6: Adversarial success rate versus adversarial smoothing factors for the TextFooler attack (on AG News).

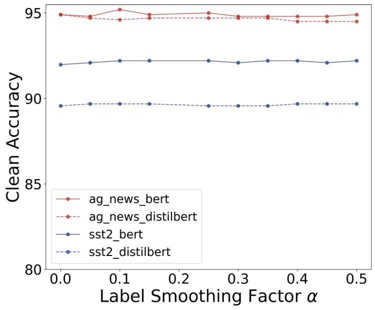

Figure 7: Clean accuracy versus label smoothing factors (on AG News and SST-2.)

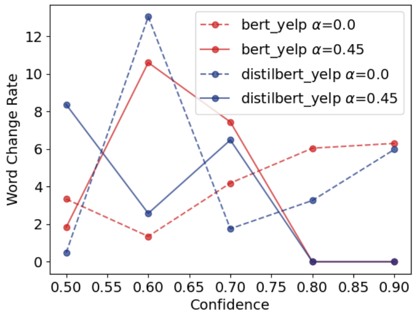

Figure 8: Average word change ratio versus confidence for in-domain inputs (No. of buckets: 10 and the number of instances in first 5 buckets [0-0.5] are 0)

Moreover, we bucket the examples based on

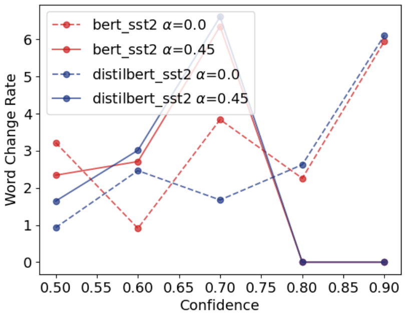

Figure 9: Average word change ratio versus confidence for out-of-domain inputs (No. of buckets: 10 and the number of instances in first 5 buckets [0-0.5] are 0)

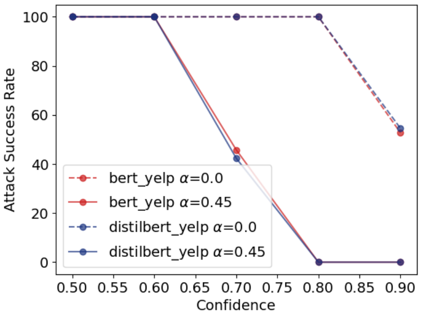

Figure 10: Adversarial success rate versus confidence for in-domain (Yelp) inputs. (Number of buckets: 10 and the number of instances in first 5 buckets [0-0.5] are 0).

the confidence scores, and plot the bucket-wise attack success rate (of the BAE attack on the Yelp dataset) versus confidence in Figure 10 and Figure 11. We observe that the label smoothing technique improves the adversarial robustness for high confidence score samples significantly. In future work, we plan to investigate the variations of robustness in label-smoothed models as a function of the model size.

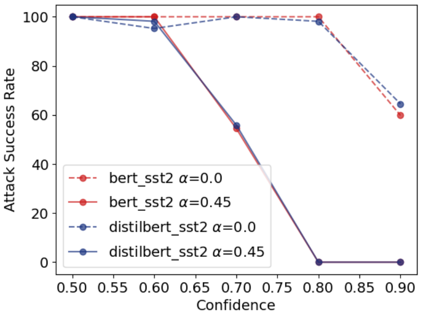

Figure 11: Adversarial success rate versus confidence for OOD inputs in the SST-2 → Yelp transfer setting.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section 5.

☑ A2. Did you discuss any potential risks of your work?
Section 6.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Yes. Abstract and section 1.

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 3, and appendix A.8.

☑ B1. Did you cite the creators of artifacts you used?
Section 3 and appendix A.8.

□ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Section 3.

□ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

□ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Appendix A.8

C ☑ Did you run computational experiments?

Section 3.

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Appendix A.8

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 3.1 and Appendix A.8.

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 3 and Appendix A.8.

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 3 and Appendix A.3, A.8.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?
Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.