# Impact of Adversarial Training on Robustness and Generalizability of Language Models

**Enes Altinisik  Hassan Sajjad[♣]  Husrev Taha Sencar**
**Safa Messaoud  Sanjay Chawla**
{ealtinisik,hsencar,smessaoud,schawla}@hbku.edu.qa
Qatar Computing Research Institute, HBKU Research Complex, Doha, Qatar

hsajjad@dal.ca
[♣]Faculty of Computer Science, Dalhousie University, Halifax, Canada

## Abstract

Adversarial training is widely acknowledged as the most effective defense against adversarial attacks. However, it is also well established that achieving both robustness and generalization in adversarially trained models involves a trade-off. The goal of this work is to provide an in depth comparison of different approaches for adversarial training in language models. Specifically, we study the effect of pre-training data augmentation as well as training time input perturbations vs. embedding space perturbations on the robustness and generalization of transformer-based language models. Our findings suggest that better robustness can be achieved by pre-training data augmentation or by training with input space perturbation. However, training with embedding space perturbation significantly improves generalization. A linguistic correlation analysis of neurons of the learned models reveal that the improved generalization is due to 'more specialized' neurons. To the best of our knowledge, this is the first work to carry out a deep qualitative analysis of different methods of generating adversarial examples in adversarial training of language models.

## 1 Introduction

Language Models (LMs) have emerged as the backbone of many tasks in AI and have extended their reach beyond NLP applications into vision and even reinforcement learning (Brown et al., 2020; Reed et al., 2022; Ramesh et al., 2022). Thus it is imperative that the generalizability and robustness of LMs be carefully assessed and evaluated.

Generalizability is the ability of a model to perform well on unseen data. Transformer-based models that are pre-trained on large unlabeled text have shown remarkable generalization ability. However, when confronted with carefully designed adversarial samples, their robustness - the ability to gracefully deal with small perturbations, suffers significantly. For example, a recent study has shown that on a classification task on a YELP data set, accuracy dropped by almost 90%, when a standard test set was replaced by an adversarial counterpart (Jin et al., 2020; Yoo and Qi, 2021; Yuan et al., 2021).

Adversarial training is a pragmatic approach to attain both generalizability and robustness. The idea is straightforward. For a given model $M$, generate adversarial samples that target $M$ and then use the samples to incrementally re-train the model. This can be done either at the pre-training or the fine-tuning stage (Liu et al., 2020).

Adversarial samples can be generated both in the input space and in the embedding space. The original work on the creation of adversarial samples for computer vision was in the input space. For example, the fast gradient sign method (FGSM) (Goodfellow et al., 2014) that perturbs a data point $x$ along the direction of the sign gradient of the loss function with respect to the input is an example of a perturbation in the input space. In the context of natural language inputs, perturbing text is challenging due to its discrete nature. Unlike continuous data, there is no systematical way to guarantee an increase in the loss function when perturbing text. For instance, if we aim to make a small modification to the word "robust" we can choose to replace a single letter within the word or substitute it with a near synonym. However, both of these perturbations may seem ad-hoc and not sufficiently principled to intentionally *increase* the loss function. Therefore, in language settings, it is often more appropriate to perform perturbations in the embedding space, where continuous representations can be manipulated in a more structured manner.

Furthermore, despite the widespread use of adversarial training to increase the robustness of models, it is not clear what their impact is on downstream tasks beyond the model's overall accuracy. For example, a deeper analysis of language models has shown that different parts of the network are responsible for different parts of speech (Belinkov

*Findings of the Association for Computational Linguistics: ACL 2023*, pages 7828–7840
July 9-14, 2023

et al., 2017; Conneau et al., 2018; Liu et al., 2019; Dalvi et al., 2022; Durrani et al., 2020). In this regard, the change in the network due to adversarial training has not yet been investigated.

Overall our contributions in this paper are threefold. Firstly, we introduce two techniques in the context of adversarial training in the embedding space, representing the regularization- and gradient-based approaches commonly used by latent space techniques. We compare these techniques using a simple one-dimensional model and hypothesize their behavior in adversarial scenarios. Secondly, we evaluate the effectiveness of input- and embedding-space adversarial training methods in terms of their generalization ability and robustness against various types of adversarial attacks in sentiment analysis. Lastly, we conduct a thorough linguistic analysis of an adversarially trained model and demonstrate that incorporating robustness through adversarial training leads to more “focused" neurons that are associated with distinct Part of Speech (POS) tags.

The rest of the paper is organized as follows. In Section 2, we discuss adversarial attacks and defenses, with a specific focus on the NLP domain. Section 3 provides a detailed explanation of embedding space adversarial techniques. In Section 4, we conduct experiments to analyze the trade-off between robustness and generalization achieved by data augmentation, input-space training, and embedding space training approaches, considering various well-known adversarial attacks. Additionally, we present our findings from linguistic correlation analysis of neurons in robust models within the same section. Finally, we finalized the paper in the concluding section.

## 2 Related Work

**Adversarial Attacks:** The purpose of an adversarial attack is to cause a model to output conflicting decisions for an input and its ‘imperceptibly’ modified version. An adversarial sample is defined as:

$$x' = x + \delta; ||\delta|| \leq \epsilon \wedge f(x, \theta) \neq f(x', \theta) \quad (1)$$

where $x'$ is the adversarial sample, $\delta$ is the perturbation added to the original data $x$, $||\delta||$ is a generic norm, $\epsilon$ is the limit of the maximum norm of the perturbation, and $f(x, \theta)$ is the output of the model parameterized by $\theta$ for input $x$. The quality of an adversarial sample is typically evaluated depending on how well $\delta$ is minimized, i.e., the minimum distortion that changes the prediction of the model on a sample.

Obtaining an exact solution for the perturbation $\delta$ is a very challenging problem. Further, even when close approximations are considered, the solution gets computationally very expensive (Szegedy et al., 2013). To solve this problem more efficiently, gradient-based methods were introduced. Accordingly, the perturbation $\delta$ is computed by taking one (Goodfellow et al., 2014) or more steps iteratively (Madry et al., 2017; Dong et al., 2018) in the direction of the gradient to maximize the loss function. Then, this high loss point is projected back onto the input space to determine the norm-bounded perturbation. In practice, projected gradient descent (PGD) approaches that, take several small steps in the direction of the gradient, are used most frequently to create strong adversarial samples (Madry et al., 2017; Papernot et al., 2016).

Other than gradient based approaches, *Jacobian-based Saliency Map Attack* (JSMA) (Papernot et al., 2016) uses the Jacobian matrix created from forward derivation of input to identify to importance of each input component to the target attack. *DeepFool* (Moosavi-Dezfooli et al., 2016), alternatively, iteratively linearizes the classifier to identify the minimum perturbation that causes a change in the classification label. *Carlini & Wagner* Attack (C&W) proposed defensive distillation strategy (Hinton et al., 2015) based approach.

**Adversarial Attacks in NLP:** Running adversarial attacks against Natural language processing (NLP) models is more challenging than widely used vision models. The discrete nature of word representations, combined with the tokenization of words into word pieces, effectively invalidates any algorithm that applies differential changes on the model input when generating an adversarial sample. Moreover, quantification of the extent to which semantic similarity and contextual relations are preserved between a text input and its modified version is not trivial.

To circumvent these limitations, many adversarial sample generation algorithms adopted the approach of substituting one or more words in the input until a misprediction occurs. The crux of this attack lies in identification of alternative words or phrases that retain the semantic intactness of the original input. For this, several methods based on word-embedding similarity (Jin et al., 2020), word synonymity (Ren et al., 2019; Zang et al., 2019), and masked language model predictions (Li et al., 2020) are proposed. However, finding appropriate word candidates may get computationally very intensive. For a sentence consisting of $m$ words with $n$ candidates to substitute each word, there are $(n+1)^m$ possible combinations to test. To perform this search efficiently, greedy search (Ren et al., 2019), genetic algorithm (Alzantot et al., 2018), and particle swarm optimization-based (PSO) (Zang et al., 2019) approaches are proposed and incorporated with word importance as determined by gradient measurements (Yoo and Qi, 2021) and word deletion (Ren et al., 2019).

An alternative approach to above substitution-based approach is applying perturbations in the embedding space directly to word embeddings. This approach avoids the expensive search step to identify the best word substitution configuration, but it requires devising a mapping from perturbed embeddings to the text domain in order to create an adversarial sample. To realize this, recent work (Yuan et al., 2021) adapted a gradient-based adversarial sample generation method to compute perturbations associated with each word embedding. Perturbed embeddings are then translated to input domain using a pre-trained masked-language modeling (MLM) head, as in (Li et al., 2020; Garg and Ramakrishnan, 2020), to create an adversarial sample that is semantically similar to the original input.

**Adversarial Defence in NLP:** The most commonly deployed method for attaining robustness against an adversarial attack is through addition of adversarial samples into the training set (Szegedy et al., 2013). This approach is known to increase model robustness in both computer vision and NLP domains. Further, it is also reported that this defence approach decreases the generalization error of a model in the absence of any attack (Yuan et al., 2021), which contradicts the commonly held opinion that there is a trade-off between generalization and robustness E: (Tsipras et al., 2019). This finding can essentially be attributed to the use of a larger training set enhanced with adversarial samples. The second approach augments the training set with newly constructed, synthetic samples. While this may seem equivalent to adding adversarial samples to the training set, data augmentation methods do not need to have an adversarial nature. Common data augmentation methods include word replacement, i.e., substituting words with their synonyms or inserting random words, random word deletions, and swapping of words between sentences (Wei and Zou, 2019). Rather than using manually-designed heuristics, the power of existing NLP models can also be harnessed for data augmentation. Reverse translation, which involves re-translation of samples from a target language back to their source language constitutes one such method that ideally preserves the semantic similarity of original and augmented samples (Edunov et al., 2018; Xie et al., 2020). The use of MLM via masking words in a sentence and replacing them with model predictions (Ng et al., 2020) is another augmentation method.

The third approach to adversarial training involves applying perturbations in the latent space (Zhu et al., 2019; Liu et al., 2020; Li and Qiu, 2021; Pan et al., 2022). This yields a simpler training procedure as it removes the need for generating adversarial samples in the input space. In (Zhu et al., 2019), a model is incrementally fine-tuned on sets of adversarially perturbed word embeddings computed after each fine-tuning step. Li et al. (2021) demonstrate that this method performs better when no constraint on the amount of perturbation is imposed. In Li and Qiu (2021), it is observed that rather than initializing the PGD step with random noise when computing perturbations for each token, using a token-dependent random noise that is fixed across all inputs is more effective. Recently, Pan et al. (2022) proposed the use of contrastive objective (Oord et al., 2018) for ensuring invariant representations by forcing the model to learn the differences between the normal input and its adversarial version.

In addition to empirical methods, certified defense methods are proposed to identify and eliminate adversarial samples. These techniques minimize misclassification within an $l_\infty$ ball bound, particularly in the vision domain (Raghunathan et al., 2018; Wong and Kolter, 2018). In the NLP domain, two main categories of certified defense methods have emerged: Interval Bound Propagation (IBP) (Jia et al., 2019; Huang et al., 2019; Shi et al., 2020) and randomized smoothing (Ye et al., 2020; Zeng et al., 2021). IBP techniques estimate the output range by iteratively applying interval constraints from the input layer to subsequent layers. However, the requirement to modify the model structure poses challenges in incorporating these methods into pre-trained models.

Randomized smoothing-based methods offer an alternative approach that is independent of the model structure. These methods utilize stochastic ensembles of input texts and leverage the statistical properties of these ensembles to offer provable robustness certification. A common approach to achieve this is by generating a few randomly modified versions of the original sample. This can be done through techniques such as random word substitutions using synonyms, as demonstrated in SAFER (Ye et al., 2020), or by employing a mask language model to substitute words, as shown in RanMASK (Zeng et al., 2021). The final prediction is then made based on the decisions made by these randomly generated samples.

Throughout the rest of the paper, we do not delve into a detailed discussion of these techniques for several reasons. Firstly, the main focus of this paper is on empirical methods and evaluating their impact. Secondly, randomized smoothing methods can be integrated into various techniques, making them applicable in different contexts. Lastly, previous findings suggest that while randomized smoothing methods demonstrate strong defense performance, they tend to underperform compared to latent space adversarial training (Li et al., 2021).

## 3 AT with Embedding Space Perturbations

Among all adversarial defenses developed for language processing models, moving the adversarial training from the input space to the embedding space offers the most advantage. This essentially allows the adoption of gradient-based adversarial training approaches that are computationally less demanding than input space methods. Although a plethora of such adversarial training methods exists, they are all essentially guided by two main principles in their approach. The first one essentially sets the training objective to minimize the loss due to worst-case perturbation induced on the training samples, instead of the average loss computed from training samples by the standard training. This group of methods essentially differ in the way they approximate the worst-case perturbation (Madry et al., 2017; Miyato et al., 2018; Zhang et al., 2019) as well as the extent and nature of perturbation applied during generation of adversarial samples (Ding et al., 2018; Wang et al., 2019; Liu et al., 2020).

The second approach primarily relies on the premise that smoothness is an important requirement of a robust model. To this objective, these methods focus on minimization of a regularized version of the loss instead of optimizing only the standard, training loss. The regularization term here ensures that there is a wide enough margin around each training data point with the decision boundary of the model through minimizing the difference between the predictions of natural and adversarial samples. Methods following this approach are distinguished based on their formulation of regularization (Szegedy et al., 2016; Zhang et al., 2019) and their coupling with the training loss described above (Gan et al., 2020; Pan et al., 2022).

In our analysis, we consider two representative methods that most effectively exemplify each approach. In practice, due to its computational efficiency, the PGD attack is most frequently used for the creation of adversarial samples. We will refer to this generic adversarial training approach as PGD-AT. The latter approach is also best characterized by the use of PGD in ensuring local distribution smoothness around natural samples. This alternative method will be referred to as LDS. We must note that improved variants of the two base methods should be expected to perform better. In this regard, robustness-generalization performance of the PGD-AT and LDS can be interpreted as lower-bounds.

The steps of both methods are presented in Algorithm 1 where the lines that differ between the two methods are highlighted as pink for PGD-AT and blue for LDS. Both methods start by randomly initializing $\delta$ with normal distribution with a mean of zero and standard deviation of $\sigma$. The loss is then calculated between the model's output of the perturbed input depending on the method, PGD-AT or LDS. The $\delta$ value is then updated by the gradient and clipped to within $\pm\epsilon$ by the projection function $\Pi$. These steps are repeated for $S$ times. The loss value is then updated by combining the standard loss with the loss associated with each method. Gradient update is then applied to model parameters.

To better examine the behavior of the two methods, we analyze a simple one-dimensional linear

**Algorithm 1** PGD-AT and LDS based adversarial training

```
Input: $E$: the number of epochs, $D = \{(x_{(i)}, y_{(i)})\}_{i=1}^{n}$: the dataset, $f(x,\theta)$: the machine learning model parametrized by $\theta$, $\delta$: the perturbation initialized by $\sigma$ and limited by $\epsilon$, $\tau$: the global learning rate, $\mu$: the adversarial learning rate, $S$: the number of PGD step, and $\Pi$ is the projection function.
for $e = 1, .., E$ do
    for $(x, y) \in \mathcal{D}$ do
        $\delta \sim \mathcal{N}(0, \sigma^2)$
        for $s = 1, .., S$ do
            $g_{adv} = \nabla_{\delta} l(f(x+\delta,\theta), y)$ %PGD-AT
            $g_{adv} = \nabla_{\delta} l(f(x,\theta), f(x+\delta,\theta))$ %LDS
            $\delta = \Pi_{||\delta|| \leq \epsilon}(\delta + \mu g_{adv})$
        end
        $g_{\theta} \leftarrow \nabla_{\theta} l(f(x,\theta), y)$
            $+\nabla_{\theta} l(f(x+\delta,\theta), y)$ %PGD-AT
        $g_{\theta} \leftarrow \nabla_{\theta} l(f(x,\theta), y)$
            $+\nabla_{\theta} l(f(x,\theta), f(x+\delta,\theta))$ %LDS
        $\theta \leftarrow \theta - \tau g_{\theta}$
    end
end
Output: $\theta$
```

| Model | Loss Function | Parameter |
|---|---|---|
| OLS | $\frac{1}{n}\sum_{i=1}^{n}(\theta.x_i - y_i)^2$ | $\theta = \frac{\sum_i x_i y_i}{\sum_i x_i^2}$ |
| PGD-AT | $\frac{1}{n}\sum_{i=1}^{n}\left\{(\theta.x_i - y_i)^2 + (\theta.(x_i+\delta) - y_i)^2\right\}$ | $\theta = \frac{\sum_i 2x_i y_i + y_i\delta}{\sum_i x_i^2 + (x_i+\delta)^2}$ |
| LDS | $\frac{1}{n}\sum_{i=1}^{n}\left\{(\theta.x_i - y_i)^2 + (\theta.(x_i+\delta) - \theta.x_i)^2\right\}$ | $\theta = \frac{\sum_i x_i y_i}{\sum_i x_i^2 + \delta^2}$ |

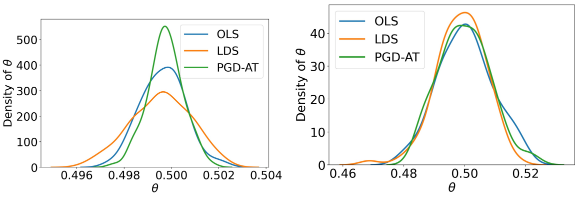
Table 1: Closed form solutions of the model parameter of a one-dimensional linear regression model under various loss functions

regression model:

$$y = \theta.x + \epsilon, \qquad \epsilon \sim N(0, \sigma^2)$$

Assuming a fixed perturbation $\delta$, we determine how the two loss functions, given in Algorithm 1, estimate the model parameter $\theta$ under noisy observations. Table 1 presents the loss functions corresponding to PGD-AT and LDS as well as the one corresponding to the standard ordinary least squares (OLS) estimation in the absence of $\delta$. The estimates for the parameter $\theta$ for the three loss functions are also given in the table (third column). Comparing PGD-AT and LDS, it can be deduced that LDS will converge to OLS only as the noise $\epsilon$ gets severe, suppressing the effect of $\delta$ in the denominator. Whereas PGD-AT can be expected to follow OLS more closely at all noise levels as $\delta$ appears both at the numerator and the denominator, thereby absorbing its effect on the estimate.

We also designed an experimental setup to test these hypotheses. A single neuron is trained based on randomly generated $(x, y)$ pairs as defined above assuming $\theta = \frac{1}{2}$ and for two different noise distributions, ($\sigma = 0.01$ and $\sigma = 0.1$) for each loss function. The models are trained for 2K epochs at a learning rate of 0.005 starting with the OLS loss. For PGD-AT and LDS models, the OLS loss is substituted by their loss function after epoch 1750 and $\delta$ values are computed as defined in Algorithm 1.



(a) (b)

Figure 1: The resulting distribution for $\theta$ values related to three different models, trained using OLS, LDS, and PGD-AT methods, when $\sigma$ is set to (a) 0.01 and (b) 0.1. A small standard deviation indicates the model's robustness and clustering around 0.5 implies better generalizability.

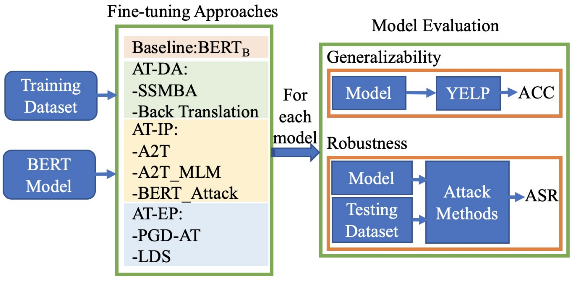


Figure 2: Evaluation pipeline of models learned using different adversarial training approaches.

The distributions of the estimated scalar model parameter $\theta$ obtained after 25 runs is displayed in Fig. 1. Essentially, the spread of the distribution signifies the robustness of a model against adversarial samples and the distribution mean relates to the generalizability of the model. In this regard, PGD-AT is seen to perform better than LDS as it yields a tighter spread in both cases. However, at higher noise levels, it can be seen that LDS provides a more accurate estimate of $\theta$. Overall, we can expect that a model trained with PGD-AT to be more robust while yielding a generalizability behavior closer to that of LDS.

## 4 Experiments

We first compare the robustness, generalization and run-time complexity of different AT strategies, following the pipeline in Fig. 2. Then, we perform a Linguistic Correlation Analysis (LCA, Dalvi et al.,

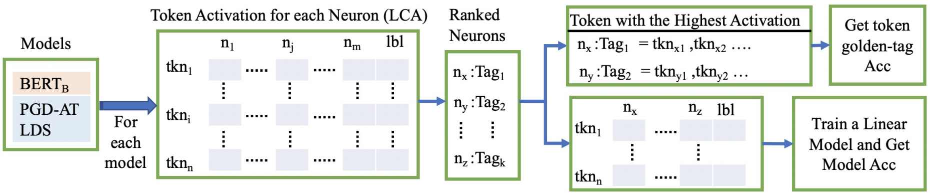

Models
BERT$_B$
PGD-AT
LDS
For each model
Token Activation for each Neuron (LCA)
$n_1$
$n_j$
$n_m$
lbl
$tkn_1$
$tkn_i$
$tkn_n$
Ranked Neurons
$n_x$ :Tag$_1$
$n_y$ :Tag$_2$
$n_z$ :Tag$_k$
Token with the Highest Activation
$n_x$ :Tag$_1$ = $tkn_{x1}$ ,$tkn_{x2}$ ….
$n_y$ :Tag$_2$ = $tkn_{y1}$ ,$tkn_{y2}$ …
Get token golden-tag Acc
$n_x$
$n_z$
lbl
$tkn_1$
$tkn_n$
Train a Linear Model and Get Model Acc

Figure 3: LCA pipeline of models learned using different adversarial training approaches.

Table 2: Robustness results. Models are evaluated using ASR (lower is better) on the MR and IMDB datasets.

| Attack | Dataset | BERT | AT-IP | | | AT-DA | | AT-EP | |
|---|---|---|---|---|---|---|---|---|---|
| | | | A2T | A2T_MLM | BERT_Attack | SSMBA | BackTranslation | LDS | PGD-AT |
| TextFooler | MR | 82.1 | 77.9 | 79.7 | 79.3 | 79.8 | **72.8** | 88.6 | 87.8 |
| | IMDB | 80.6 | 86.5 | 65.3 | 72.8 | 81.3 | **45.9** | 91.5 | 94.7 |
| A2T | MR | 33.8 | 27.6 | 30.8 | 26.5 | 34.2 | 30.4 | 22.3 | **20.9** |
| | IMDB | 59.5 | 51.1 | 43.7 | 49.4 | 59.2 | **36.4** | 56.9 | 43.0 |
| BAE | MR | 52.1 | 44.1 | 45.5 | **44.0** | 49.2 | 47.1 | 55.3 | 52.9 |
| | IMDB | 68.8 | 65.0 | 52.5 | 57.9 | 61.5 | **41.4** | 66.5 | 61.0 |
| PSO | MR | 79.8 | 75.0 | **72.7** | 74.7 | 78.1 | 75.6 | 79.8 | 80.7 |
| | IMDB | 46.4 | 35.3 | 35.4 | **30.2** | 41.8 | 42.8 | 70.8 | 66.2 |
| **Average** | MR | 62.0 | **56.1** | 57.2 | **56.1** | 60.3 | 56.5 | 61.5 | 60.5 |
| | IMDB | 63.8 | 59.5 | 49.2 | 52.6 | 61.0 | **41.6** | 71.4 | 66.2 |

2019) as implemented in the NeuroX toolkit (Dalvi et al., 2023) to gain better insights into the dynamics of the learned models, as illustrated in Fig. 3.

**Baselines:** We compare standard BERT (Devlin et al., 2018) with seven versions of adversarially trained BERT models using methods from three families of AT approaches: (1) AT with pre-training data augmentation (AT-DA), (2) AT with input space perturbations (AT-IP) and (3) AT with embedding space perturbations (AT-EP), on the task of sentiment classification. Specifically, for AT-DA, we experiment with SSMBA (Ng et al., 2020) and BackTranslation (Xie et al., 2020). For AT-IP, we use A2T, A2T_MLM (Yoo and Qi, 2021) and BERT_attack (Li et al., 2020). For AT-EP, we report results on LDS (Szegedy et al., 2016; Zhang et al., 2019) and PGD-AT (Gan et al., 2020; Pan et al., 2022).

**Datasets**: We fine-tune all models on the Internet Movie Database (IMDB, Maas et al., 2011) and Movie Reviews (MR, Pang and Lee, 2005) datasets and test on the corresponding testing splits, as well as on YELP dataset (Zhang et al., 2015) for out-of-distribution assessment of the models.

**Attack methods:** We assess the robustness of the models under four different attacks which replace words in the input space using different strategies. (1) TextFooler (Jin et al., 2020) first searches for the word that results in the highest change in the sentiment score, when removed, then replaces it with the nearest neighbouring word in the embedding space. (2) BAE (Garg and Ramakrishnan, 2020) masks a portion of the text and using a BERT masked language model to generate alternatives for the masked words. (3) A2T (Yoo and Qi, 2021) selects the word with the largest loss gradient w.r.t its embedding and replaces it with a synonym generated from a counterfitted word embedding (Mrkšić et al., 2016). (4) PSO (Zang et al., 2019) uses sememe-based word substitution and particle swarm optimization-based search algorithm to find good adversarial examples for a given input text.

**Evaluation metrics:** we assess (1) generalization via computing the accuracy values on in-distribution and out-of-distribution datasets, (2) robustness using the Attack Success Rate (ASR) representing the ratio of the number of successful attacks to the number of samples, as well as (3) the time complexity measured via the fine-tuning run-time of the BERT model over 4 epochs.

**Implementation details:** For AT-DA and AT-IP methods, we use the parameters proposed by the corresponding papers. For our PGD-AT and LDS approaches, we limit the number of PGD steps to 3 and the perturbations L2-norm to 0.003. All experiments are conducted on Nvidia v100 Tensor Core GPU.

**Run-time results:** We report the time for fine-tuning the models over 4 epochs in Tab. 3. The AT-DA approaches results in the shortest fine-tuning time as adversarial examples are generated once for every sample before the training, unlike in AT-IP and AT-EP where adversarial examples are generated at every training iteration. AT-EP methods, are around 1.5 times slower to fine-tune than the standard BERT model as generating the adversarial examples requires an additional backward pass for computing the gradient of the loss, at every training iteration. As expected, AT-IP methods are the most time consuming as they involve a combinatorial search over a large number of input space configurations. For example, the fastest approach in this class, A2T, needs 6 seconds for a single adversarial example generation, which is around 10 times slower than the other approaches.

Table 3: Run-time results. We report the fine-tuning runtime over 4 episodes on the MR and IMDB datasets.

| | Models | Run Time (in min) | |
|---|---|---|---|
| | | IMDB | MR |
| | BERT | **79.0** | **38.2** |
| AT-DA | SSMBA | 112.8 | 46.4 |
| | BackTranslation | 210.5 | 66.0 |
| AT-IP | A2T | 1600.5 | 448.5 |
| | A2T_MLM | 1494.3 | 504.7 |
| | BERT_Attack | 1495.2 | 461.5 |
| AT-EP | LDS | 163.4 | 64.2 |
| | PGD-AT | 158.2 | 69.0 |

**Robustness results** are shown in Tab. 2. The lower the ASR the better is the model in withstanding the attack. As expected, the most effective methods against adversarial attacks are the AT-IP ones. This is due to the fact that the only class of approaches were it's possible to match the attack and the defense strategies, i.e., train on perturbations generated from the attack strategies, is AT-IP, as attacks in language models operate in the input space. Among AT-AD methods, BackTranslation is the most robust method on the IMDB dataset. We found that this is due to IMDB having in average long sentences which makes it easier to generate good and diverse adversarial examples to train on, via back translation. Our results show that AT-EP methods are the least robust. In particles, LDS-AT struggle in the sentiment classification task due to noisy ground-truth label, i.e., sentiments are mostly not binary but the ground truth labels are.

**Generalization results** are reported in Tab. 4. AT-DA accuracy values are comparable to BERT. Hence, it looks like AT-DA generalization capabilities are not traded-off for better robustness as it is the case of AT-IP approaches. This is due to the fact that adversarial examples from SSMBA (self-supervised-based) and BackTranslation (translation-based) are generated while taking the global context into account. So they are unlikely to change the semantics of the input text and hence the decision boundaries. These methods are however unpractical for usage inside of the training loop. More efficient techniques, e.g., based on local search in the embedding space, are used by AT-IP methods. This however might not always lead to preserving the semantics of the original input text, which also means that assigning the label of the ground truth input to these adversarial examples might be inappropriate or noisy. Such hard examples are well known to encourage overfitting and hence reduce the generalization ability of the model. This explains the significant drop in both in and out-of-distribution accuracy values of AT-IP approaches. The best generalization results are obtained using AT-EP methods. We notice that PGD-AT consistently improves upon BERT. This phenomena doesn't occur in vision where generalization is well know to drop in adversarially trained models. To the best of our knowledge, we are the first to report this in language models trained with embedding space perturbation. In order to gain a better understanding of the reasons behind this phenomena, we investigate the learned dynamics of deepnets trained with AT-EP methods using Linguistic Correlation Analysis (next paragraph). Specifically, we want to validate that the achieved accuracy was due to better learning to solve of the task at hand and not just due to memorizing the training data.

**Linguistic Correlation Analysis (LCA, Dalvi et al., 2019)** is used to identify the most salient neurons for a given linguistic property like a Parts-of-Speech (POS) tag (Sajjad et al., 2022). To achieve this, we first match words to neurons, then assess if the matched words have the linguistic property of interest. As the sentiment prediction task is not appropriate for word level analysis, i.e., same words can be part of different sentiment classes, we focus on POS tagging task. We fine-tune BERT models using AT-EP methods on the publicly available Penn Treebank dataset (Marcinkiewicz, 1994). We use LCA to generate a list of the top-5 firing neu-

Table 4: Generalization results. We report the accuracy values on IMDB/MR (in-distribution) and YELP (out-of-distribution) datasets for BERT models fine-tuned on IMDB/MR for the task of sentiment classification.

| | Models | IMDB | | MR | |
|---|---|---|---|---|---|
| | | IMDB | YELP | MR | YELP |
| | BERT | 93.49 | 91.24 | 85.27 | 87.06 |
| AT-DA | SSMBA | 93.49 | 91.17 | 85.24 | 87.72 |
| | BackTranslation | 93.44 | 91.50 | 84.96 | 87.77 |
| AT-IP | A2T | 92.59 | 89.97 | 83.58 | 83.62 |
| | A2T_MLM | 92.70 | 89.15 | 83.90 | 81.79 |
| | BERT_Attack | 92.63 | 90.04 | 84.61 | 80.41 |
| AT-EP | LDS | 93.24 | **92.09** | 86.49 | 81.80 |
| | PGD-AT | **93.80** | **92.11** | **86.59** | **88.16** |

Table 5: LCA results. The association strength between POS tags and neurons.

| POS | BERT | | | LDS | | | PGD-AT | | |
|---|---|---|---|---|---|---|---|---|---|
| | Match | Total | % | Match | Total | % | Match | Total | % |
| JJ | 2 | 15 | 13.33 | 2 | 15 | 13.33 | **6** | 10 | **60.00** |
| JJR | 3 | 9 | 33.33 | 4 | 9 | 44.44 | **7** | 13 | **53.84** |
| MD | 0 | 5 | 0.00 | **3** | 5 | **60.00** | 2 | 5 | 40.00 |
| VBD | **5** | 5 | **100.00** | 0 | 5 | 0.00 | 0 | 5 | 0.00 |
| : | 0 | 5 | 0.00 | 1 | 5 | 20.00 | **3** | 5 | **60.00** |
| VBZ | 4 | 10 | 40.00 | 7 | 9 | 77.77 | **9** | 10 | **90.00** |
| RB | **9** | 10 | **90.00** | **9** | 10 | **90.00** | 6 | 10 | 60.00 |
| VBG | 10 | 12 | 83.33 | 14 | 18 | 77.77 | **15** | 15 | **100.00** |

rons for every POS tag and leverage these lists to perform two types of analysis: (1) neurons-POS tags association strength analysis and a (2) a neural ablation analysis. To assess the neurons-tag association strength, given the list of the top-firing neurons from LCA, we next generate a list of the words in the testing data with the highest activation values for these neurons. Then, we compute the intersection between the generated word list and the ground-truth one, i.e., the list of words with label being the POS tag of interest in the testing data. A large intersection set means that the neurons learned to specialize in predicting specific POS tags, i.e., they learned the linguistic nuances of the task and are unlikely to have just memorized the training data. Results in Tab. 5[1] show that our AT-EP learn more 'focused' neurons as measured by the intersection ratio (match/total). In particular, PGD-AT significantly improves upon the standard $\text{BERT}_\text{B}$ model.

Table 6 provides words corresponding to select POS tags obtained from the models trained with the $BERT_B$, the LDS, and PGD-AT methods. For the second analysis, i.e., the neural ablation study, we create a linear regression model using only activations of the top 10 ranked neurons. Results are shown in Tab. 7. PGD-AT and LDA achieve a significantly higher performance than BERT, which further support the observation that AT helped better learn the intricacies of the tasks and explains the improvement of the generalization abilities of the AT-EP approaches (e.g., in Tab. 4).

## 5 Conclusions

In this paper we have carried out an extensive study of adversarial training methods (ATMs) to understand their impact on robustness and generalizability for transformer-based deep language models. We can draw the following conclusions from our study. First, non-adversarial data augmentation improves both generalization and robustness over the baseline BERT model. Adversarial training in the input space yields better robustness compared to both non-adversarial data augmentation and embedding space adversarial training. In contrast, adversarial training in the embedding space exhibits best generalization behavior. Among PGD-AT and LDS methods, our results show that the PGD-AT is consistently more robust and generalizable. Overall, our results show that unlike in computer vision domain where gradient-based adversarial training yields the best robustness and generalization trade-off, for language processing models input-space training methods are indispensable.

For future work we will consider combining data augmentation, input-space training, and embedding space training approaches together. We would also like to extend our theoretical understanding of the trade-off between robustness and generalizability for language models. In connection, the impact of ATMs for other downstream applications needs to be studied.

## Limitations

All our experiments are performed using the BERT-small language model due to the computational requirements of generating and testing models considering many configurations of adversarial training and attack methods. Although using larger language models might have provided different performance measurements, our findings that compare input- and embedding-space adversarial training

[1] Definitions of POS tags with their order in the table: adjective; adjective, comparative; modal; verb, past tense; colon, semi-colon; verb, 3rd person singular present; adverb; verb, gerund or present participle

Table 6: Examples of the most related words for different POS tags for models trained with the $BERT_B$, the LDS, and PGD-AT methods. The words are bolded when their actual tags match with the associated tag, where the actual tags correspond to the most frequent tags of the words based on the POS-tagged training data.

| POS | $BERT_B$ | LDS | PGD-AT |
|---|---|---|---|
| VBZ | **indicates** teenage And **begins** **reflects explains** evil Previously automatic reckless | **indicates denies erodes explains** **resembles** And **runs** **adds** trains | **indicates accounts refuses agrees** **is has believes** And **adds begins** |
| JJ | Rae away **little** Springs Nelson live **equal** What explain Giants Who Aktiebolaget skyrocketed what rung | Aktiebolaget least plummeted Do policies **little** told What **equal** securities Dallara added said most cardboard | **bright** away what **high** **strong cold** skyrocketed **green** What **same** |
| JJR | **newer** meaning **greater** punish included banking close **smaller** her | included **newer greater smaller** indicated shipbuilding arranged **Higher** her | included **newer stronger** meaning **smaller greater** indicated planning **higher** close **Higher lower** least |
| MD | associated bright required severe denied | apart **shall might must** fallen | fallen **shall** expected **might** apart |
| VBD | **restored bothered notched mixed began** | expire face exist become buy | expire face become exist disagree |

Table 7: LCA results. Neural ablation study.

| BERT | LDS | PGD-AT |
|---|---|---|
| 34.2% | 38.6% | 35.3% |

methods are expected to remain unchanged. Another limitation of our work is the semantic gap between attacks in input and embedding space needs further research. Specifically, how do perturbations in the embedding space get translated in the input space? Finally, other forms of robustness techniques, besides adversarial training, in the context of large language models require examination.

## Ethics Statement

The work studied the impact of several adversarial training methods on robustness and generalization. The work did not result in any new dataset and model and it has no potential ethical issues. On the positive side, the work targets two important attributes of trustworthy AI i.e. robustness and generalization. Our work provides an insightful comparison of the input-space and embedding space adversarial training approaches and will positively impact the future research work in this area.

## 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*.

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017. What do Neural Machine Translation Models Learn about Morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, Vancouver. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Fahim Dalvi, Nadir Durrani, Hassan Sajjad, Yonatan Belinkov, D. Anthony Bau, and James Glass. 2019. What is one grain of sand in the desert? analyzing individual neurons in deep nlp models. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI, Oral presentation)*.

Fahim Dalvi, Abdul Rafae Khan, Firoj Alam, Nadir Durrani, Jia Xu, and Hassan Sajjad. 2022. Discovering latent concepts learned in BERT. In *International Conference on Learning Representations*.

Fahim Dalvi, Hassan Sajjad, and Nadir Durrani. 2023. Neurox library for neuron analysis of deep nlp models. In *Proceedings of the Association for Computational Linguistics (ACL)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Gavin Weiguang Ding, Yash Sharma, Kry Yik Chau Lui, and Ruitong Huang. 2018. Mma training: Direct input space margin maximization through adversarial training. *arXiv preprint arXiv:1812.02637*.

Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. 2018. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193.

Nadir Durrani, Hassan Sajjad, Fahim Dalvi, and Yonatan Belinkov. 2020. Analyzing individual neurons in pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4865–4880, Online. Association for Computational Linguistics.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. *arXiv preprint arXiv:1808.09381*.

Zhe Gan, Yen-Chun Chen, Linjie Li, Chen Zhu, Yu Cheng, and Jingjing Liu. 2020. Large-scale adversarial training for vision-and-language representation learning. *Advances in Neural Information Processing Systems*, 33:6616–6628.

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

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

Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7).

Po-Sen Huang, Robert Stanforth, Johannes Welbl, Chris Dyer, Dani Yogatama, Sven Gowal, Krishnamurthy Dvijotham, and Pushmeet Kohli. 2019. Achieving verified robustness to symbol substitutions via interval bound propagation. *arXiv preprint arXiv:1909.01492*.

Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. 2019. Certified robustness to adversarial word substitutions. *arXiv preprint arXiv:1909.00986*.

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 the AAAI conference on artificial intelligence*, volume 34, pages 8018–8025.

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*.

Linyang Li and Xipeng Qiu. 2021. Token-aware virtual adversarial training in natural language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8410–8418.

Zongyi Li, Jianhan Xu, Jiehang Zeng, Linyang Li, Xiaoqing Zheng, Qi Zhang, Kai-Wei Chang, and Cho-Jui Hsieh. 2021. Searching for an effective defender: Benchmarking defense against adversarial word substitution. *arXiv preprint arXiv:2108.12777*.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019. Linguistic knowledge and transferability of contextual representations. 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 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics.

Xiaodong Liu, Hao Cheng, Pengcheng He, Weizhu Chen, Yu Wang, Hoifung Poon, and Jianfeng Gao. 2020. Adversarial training for large neural language models. *arXiv preprint arXiv:2004.08994*.

Andrew 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 the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150.

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*.

Mary Ann Marcinkiewicz. 1994. Building a large annotated corpus of english: The penn treebank. *Using Large Corpora*, 273.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. 2018. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. 2016. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582.

Nikola Mrkšić, Diarmuid O Séaghdha, Blaise Thomson, Milica Gašić, Lina Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting word vectors to linguistic constraints. *arXiv preprint arXiv:1603.00892*.

Nathan Ng, Kyunghyun Cho, and Marzyeh Ghassemi. 2020. Ssmba: Self-supervised manifold based data augmentation for improving out-of-domain robustness. *arXiv preprint arXiv:2009.10195*.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Lin Pan, Chung-Wei Hang, Avirup Sil, and Saloni Potdar. 2022. Improved text classification via contrastive adversarial training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11130–11138.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. *arXiv preprint cs/0506075*.

Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. 2016. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, pages 372–387. IEEE.

Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. 2018. Certified defenses against adversarial examples. In *International Conference on Learning Representations*.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*.

Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. 2022. A generalist agent. *arXiv preprint arXiv:2205.06175*.

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*, pages 1085–1097.

Hassan Sajjad, Nadir Durrani, and Fahim Dalvi. 2022. Neuron-level interpretation of deep NLP models: A survey. *Transactions of the Association for Computational Linguistics*, 10:1285–1303.

Zhouxing Shi, Huan Zhang, Kai-Wei Chang, Minlie Huang, and Cho-Jui Hsieh. 2020. Robustness verification for transformers. In *International Conference on Learning Representations*.

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*.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. 2019. Robustness may be at odds with accuracy. In *International Conference on Learning Representations*.

Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. 2019. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations*.

Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*.

Eric Wong and Zico Kolter. 2018. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International conference on machine learning*, pages 5286–5295. PMLR.

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. 2020. Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems*, 33:6256–6268.

Mao Ye, Chengyue Gong, and Qiang Liu. 2020. Safer: A structure-free approach for certified robustness to adversarial word substitutions. *arXiv preprint arXiv:2005.14424*.

Jin Yong Yoo and Yanjun Qi. 2021. Towards improving adversarial training of nlp models. *arXiv preprint arXiv:2109.00544*.

Lifan Yuan, Yichi Zhang, Yangyi Chen, and Wei Wei. 2021. Bridge the gap between cv and nlp! a gradient-based textual adversarial attack framework. *arXiv preprint arXiv:2110.15317*.

Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun. 2019. Word-level textual adversarial attacking as combinatorial optimization. *arXiv preprint arXiv:1910.12196*.

Jiehang Zeng, Xiaoqing Zheng, Jianhan Xu, Linyang Li, Liping Yuan, and Xuanjing Huang. 2021. Certified robustness to text adversarial attacks by randomized [mask]. *arXiv preprint arXiv:2105.03743*.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. 2019. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pages 7472–7482. PMLR.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. 2019. Freelb: Enhanced adversarial training for language understanding.

## ACL 2023 Responsible NLP Checklist

### A For every submission:

☑ A1. Did you describe the limitations of your work?
*Under the Limitation*

☑ A2. Did you discuss any potential risks of your work?
*Under the Ethics Statement*

☑ A3. Do the abstract and introduction summarize the paper's main claims?
*Section 1*

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

### B ☑ Did you use or create scientific artifacts?

*4*

☑ B1. Did you cite the creators of artifacts you used?
*4*

□ 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)?
*4*

☒ 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?
*All datasets are publicly available*

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
*4*

□ 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.
*Not applicable. All datasets are publicly available*

### C ☑ Did you run computational experiments?

*4*

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
*4*

*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?
*4*

☑ 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?
*4*

☑ 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.)?
*4*

**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.*