Similarizing the Influence of Words with Contrastive Learning to Defend Word-level Adversarial Text Attack

Pengwei Zhan‡, Jing Yang§, He Wang§, Chao Zheng§, Xiao Huang§, Liming Wang§
§Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
‡School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
{zhanpengwei,yangjing,wanghe6029}@iie.ac.cn
{zhengchao1135,huangxiao,wangling}@iie.ac.cn

Abstract

Neural language models are vulnerable to word-level adversarial text attacks, which generate adversarial examples by directly substituting discrete input words. Previous search methods for word-level attacks assume that the information in the important words is more influential on prediction than unimportant words. In this paper, motivated by this assumption, we propose a self-supervised regularization method for Similarizing the Influence of Words with Contrastive Learning (SIWCon) that encourages the model to learn sentence representations in which words of varying importance have a more uniform influence on prediction. Experiments show that SIWCon is compatible with various training methods and effectively improves model robustness against various unforeseen adversarial attacks. The effectiveness of SIWCon is also intuitively shown through qualitative analysis and visualization of the loss landscape, sentence representation, and changes in model confidence.

1 Introduction

Neural language models have achieved impressive performance in various natural language processing (NLP) tasks, but they are also proven vulnerable to adversarial examples, which induce incorrect model output by adding small perturbations to natural inputs (Szegedy et al., 2014; Jia and Liang, 2017). Unlike attacks on images, which are performed by directly adding imperceptible continuous noise to the input, adversarial text attacks are commonly performed by substituting input text due to the discrete and non-differentiable nature of text (Gao et al., 2018; Alzantot et al., 2018; Li et al., 2019; Zhan et al., 2022b; Garg and Ramakrishnan, 2020). Among the various granularities of adversarial text attacks, word-level attacks have been more focused on by recent works for their effectiveness in maintaining semantic similarity and grammatical correctness. Unlike character-level and sentence-level attacks, word-level attacks are less likely to be detected by spell checkers or to undermine the overall coherence of a sentence (Ebrahimi et al., 2018; Iyyer et al., 2018; Liang et al., 2018).

Under a unified framework, word-level attacks can always be formulated as a combinatorial optimization problem (Yoo et al., 2020; Morris et al., 2020a,b), and various attack methods can be decomposed into Search Space and Search Method. The search space contains the possible substitutions for each word, while the search method determines the substitution order and strategy for selecting the optimal substitution from the search space. Since the search space may be model-agnostic, we should focus on the search method for the potential of improving the robustness against word-level attacks.

Previous search methods for word-level attacks are based on the assumption: different words in

Figure 1: The motivation of SIWCon. The normally trained model considers the information in important words to have a significant impact on prediction, and as a result, search methods that prioritize substituting important words are more likely to find adversarial examples. SIWCon, on the other hand, considers the information in both important and unimportant words to have a similar degree of influence on prediction, thus making it less possible for search methods that focus on important words to find optimal substitutions.
a sentence contribute differently to model prediction, with the information in important words being more influential than the information in unimportant words. Therefore, following the word importance scores obtained through attribution methods, the attack can be seen as a process of iteratively substituting words in a sentence, with important words substituted first, followed by unimportant words. For example, the search methods Word Importance Ranking (WIR) (Gao et al., 2018; Jin et al., 2020; Li et al., 2020) and PWWS (Ren et al., 2019) obtain word importance using Occlusion (Zeiler and Fergus, 2014), then WIR performs substitution in descending order of word importance and PWWS formulates token scores that use word importance as weights to guide the attack.

Following this assumption, the success of word-level attacks can be explained. The words in a sentence can be classified as important words, which contain more influential information for prediction, or unimportant words, which contain less influential information. Search methods that substitute important words first can perturb more influential information in each attack step, making the model more likely to be deceived. Therefore, it is natural to wonder: will the model be more robust when the information in both important and unimportant words has a similar degree of influence on prediction? Motivated by this question, we propose a self-supervised regularization method for Similarizing the Influence of Words with Contrastive Learning (SIWCon) that improves the model robustness against word-level attacks. The motivation of our method is illustrated in Figure 1. We summarize our main contributions as follows:

1. We discuss the relationship between model robustness and the influence of information in words of different importance.

2. We propose SIWCon, a contrastive learning method that improves the robustness of language models by encouraging models to learn sentence representations that consider the information in words of different importance to have a more similar influence on prediction.

3. We evaluate SIWCon against several attack methods on three models of different architectures and on Movie Review (MR), SST2, and IMDB datasets. Results show that SIWCon improves the model robustness against unforeseen adversarial attacks without learning from any adversarial perturbation.

4. We provide qualitative analysis and visualization on loss landscape, sentence representation, and model confidence change, intuitively showing the effectiveness of SIWCon.

2 Related Works

Robustness of Language Models. The current methods for improving the robustness of language models ignore the assumption discussed in §1. While some works attempt to detect or transform potential adversarial examples in the training set (Jin et al., 2022a; Gao et al., 2021), some previous works construct the pair examples (Zhou et al., 2019; Mozes et al., 2021), this does not actually improve the model’s robustness. Other methods, such as performing certifiably robust training through interval bound propagation (IBP), can be computationally costly and difficult to scale to large models like BERT (Jia et al., 2019; Huang et al., 2019). Additionally, it has been reported that while IBP improves adversarial accuracy, it comes at the huge cost of reduced clean accuracy (Wang et al., 2021). Some works try to perform adversarial training by incorporating adversarial examples in the training set (Jin et al., 2020; Li et al., 2021), while this method can only improve the robustness against the adversarial perturbations that the model has seen. Moreover, generating adversarial examples is time-consuming, thus adversarial training is difficult to scale to a large dataset. In this paper, based on the ignored assumption, we discuss the model robustness from new perspectives, focusing on attribution and sentence representation.

Contrastive Learning. Contrastive learning is first proposed in computer vision tasks to help models learn better image representation (Chen et al., 2020a,b; He et al., 2020; Pan et al., 2021). This self-supervised learning method alleviates the dependence on the costly labeled data. Recently, encouraged by the superior performance, various contrastive learning methods have been proposed for NLP tasks. Following the discrete nature of text, some previous works construct the pair examples by augmenting the input sentence (Giorgi et al., 2021; Wu et al., 2020; Fang and Xie, 2020; Zhan et al., 2022a; Gao et al., 2021), e.g., by word deleting, reordering, substituting, and back-translating, or by augmenting the word embedding (Yan et al., 2021), e.g., by shuffling, cutting off, dropping out the embedding matrix. Unlike the previous works.
that aim to improve the downstream performance, we focus on improving the model robustness.

3 Methodology

3.1 Preliminaries

Suppose we have the input text $X \in \mathcal{X}$ and the output labels $Y \in \mathcal{Y} = \{1, \ldots, C\}$ that follow the data distribution $\mathcal{D}$. A model $f_\theta : \mathcal{X} \rightarrow \mathcal{Y}$ that maps the input text to the output probability space is trained by minimizing $L_{ce}(X, Y; \theta)$:

$$
E_{(X,Y) \sim \mathcal{D}}[−\log \frac{\exp(w^T_{true}r_\theta(X))}{\sum_{k=1}^C \exp(w^T_k r_\theta(X))}],
$$

(1)

where $w_Y \in \mathcal{W}$ denotes the model classification parameters toward class $Y$, $\mathcal{W}$ is the overall classification parameters, and $r_\theta(\cdot)$ denotes the latent model classification encoded by the model $f$ with parameters $\theta$. The well-trained model can learn the distribution of data and predict the input sentence based on the posterior probability:

$$
P(Y_{true}|X) = \frac{\exp(w^T_{true}r_\theta(X))}{\sum_{k=1}^C \exp(w^T_k r_\theta(X))},
$$

(2)

where $w_{true}$ denotes the classification parameters toward the ground-truth class $Y_{true}$. To attribute the prediction $P(Y_{true}|X)$, i.e., identifying the words that are most influential on the prediction (Li et al., 2016b; Ross et al., 2017; Sundararajan et al., 2017; Kim et al., 2020), we use the gradient-based attribution method (Feng et al., 2018; Li et al., 2016a; Arras et al., 2016; Situ et al., 2021). The influence score of word $x_i \in X$ can be formally defined as:

$$
\text{Score}(x_i) = \frac{\partial w^T_{true}r_\theta(X)}{\partial \text{emb}(x_i)},
$$

(3)

where $\text{emb}(\cdot)$ denotes embedding, and $\|\cdot\|_2$ denotes $L^2$ norm. The influence of a word is the norm of the influence score of every embedding dimension.

3.2 Word-level Adversarial Attack

Following the analysis of word-level adversarial attacks in §1, an adversarial example $X^{adv}$ generated by search methods from a normal example $X = (x_n)_{n \in \{1, \ldots, N\}}$ can be formulated as:

$$X^{adv} = O(X) = o(x_n)_{n \in \{1, \ldots, N\}},$$

s.t. $\forall n \in \{1, \ldots, N\}$, $\Delta x_n < \delta$,

and $\Delta X < \varepsilon$,

(4)

and $\arg \max_{Y \in \mathcal{Y}} P(Y|X^{adv}) \neq \arg \max_{Y \in \mathcal{Y}} P(Y|X)$, where $O(X)$ denotes performing word-level substitution on sentence $X$, $o(x_n)$ denotes substituting the word $x_n$ with a new word from a finite search space that contains all qualified substitutions, if possible. $\Delta x_n$ and $\delta$ respectively denote the difference and the maximum allowed difference between $x_n$ and $o(x_n)$, $\Delta X$ and $\varepsilon$ respectively denote the difference and the maximum allowed difference between $X$ and $O(X)$. $\delta$ and $\varepsilon$ are used to filter qualified substitutions in the search space, which may mainly focus on the semantics and the $L^p$ norm of the embedding distance of each word and the entire sentence, ensuring the adversarial example is imperceptible to humans.

To generate adversarial examples more effectively, the search methods of current attacks, i.e., the strategies to perform $o(\cdot)$, follow the assumption that the information in important words is more influential than the information in unimportant words, and heavily rely on attribution results like (3). These methods attempt to substitute important words first to perturb more influential information in each attack step. Therefore, if different words in a sentence have a similar slight influence on prediction, the attacks should only slightly impact the model prediction in each attack step. To this end, we detail the SIWCon regularization method next.

3.3 The SIWCon Regularization

Recall that the goal of SIWCon is to **similarize the influence of words**. After regularization, the influence of different words on prediction should be similarly slight. To formally define this goal, we first define the 40% of words in a sentence with the highest and lowest influence scores as the important and unimportant words, respectively, following the attribution results of (3). We then propose two efficient non-deterministic data augmentation operations, $t^{imp}(\cdot)$ and $t^{ump}(\cdot)$, which respectively means randomly removing important and unimportant words in a sentence. Therefore, under the training scenario of (1), the primary goal of SIWCon can now be formulated as:

$$
\min_{\theta} \|Q_{imp.} - Q_{ump.}\| : \begin{align*}
Q_{imp.} &= \mathbb{E}_{(X,Y) \sim \mathcal{D}}[P(Y_{true}|X) - P(Y_{true}|X^{imp})], \\
Q_{ump.} &= \mathbb{E}_{(X,Y) \sim \mathcal{D}}[P(Y_{true}|X) - P(Y_{true}|X^{ump})],
\end{align*}
$$

(5)

where $X^{imp}$ is an augmentation sampled from $t^{imp}(X)$, and $X^{ump}$ is an augmentation sampled...
from $t^{\text{imp}}(X)$. $Q_{\text{imp}}$ and $Q_{\text{ump}}$ measure the extent of model confidence decrease when a random part of information in the important and unimportant words is lost, indicating the overall influence of the information in words of different importance on prediction. The complete objective of SIWCon can be further decomposed into two perspectives:

**Objective 1:** The influence of different words should be similar, thus the model should treat the sentences with information in words of different importance lost ($X^{\text{imp}}$ and $X^{\text{ump}}$) similarly.

**Objective 2:** The influence of different words should be slight, thus the model should treat the sentences with different information lost ($X^{\text{imp}}$ and $X^{\text{ump}}$) similarly to the original sentence that contains complete information ($X$).

To achieve Objective 1 and Objective 2, and further the goal of SIWCon, we use a contrastive loss objective from the perspective of sentence representation. To define the contrastive loss objective, for convenience, we first define the calculation $S$:

$$S_{(i,j)}^{(k,l)} = \exp(\sin[r_\theta(X^k_i), r_\theta(X^l_j)]/\tau),$$

where $k, l \in \{\text{imp, ump, } \cdot \}$, respectively indicate the augmentation sampled from $t^{\text{imp}}(\cdot)$, the augmentation sampled from $t^{\text{ump}}(\cdot)$, and the normal example, $i, j$ are the example indices, $\sin[r_i, r_j] = r_i^T r_j/\|r_i\|\|r_j\|$ is the cosine similarity, $\tau$ is a temperature parameter similar to the NT-Xent loss (Chen et al., 2020a; van den Oord et al., 2018). Then the contrastive loss function for an example in a mini-batch $X_i \in \{X\}_i^{B_{i=1}}$ is defined as:

$$\mathcal{L}_{\text{SIWCon}}(X_i; \theta) = \mathbb{E}_{X_i} \left[ -\log \frac{S_{\text{positive}}}{\sum_{j=1}^{B_{i=1}} (S_{\text{negative}})} \right],$$

where

$$S_{\text{positive}} = S_{(i,i)}^{(\text{imp, imp})} + S_{(i,i)}^{(\text{ump, imp})} + S_{(i,i)}^{(\text{ump, ump})},$$

$$S_{\text{negative}} = S_{(i,j)}^{(\cdot, \cdot)} + \mathbbm{1}_{[i \neq j]}[S_{(i,j)}^{(\text{ump, imp})} + S_{(i,j)}^{(\text{ump, ump})}],$$

$B$ is the batch size, $\mathbbm{1}[\cdot]$ is an indicator function that equals 1 if the condition $[\cdot]$ is true; otherwise, it equals 0. Specifically, to calculate the loss for each mini-batch, we first randomly sample the augmentations $X_i^{\text{imp}}$ from $t^{\text{imp}}(X_i)$ and the augmentations $X_i^{\text{ump}}$ from $t^{\text{ump}}(X_i)$ for each example in the mini-batch. The general framework of SIWCon is shown in Figure 2.

To achieve Objective 1, we use the term $S_{(i,i)}^{(\text{imp, imp})}$ in the numerator. This constraint maximizes the similarity between the representations of the augmentations with important and unimportant words removed, making the different degrees of incomplete information in the augmentations have a similar impact on the prediction.

To achieve Objective 2, we use the term $S_{(i,j)}^{(\cdot, \cdot)}$ and $S_{(i,i)}^{(\cdot, \cdot)}$ in the numerator. These constraints maximize the similarity between the original sentence and the two augmentations, making the incomplete information in the remaining words of the augmentations have a similar influence as the complete information in the normal sentence.

Intuitively, the semantics of different examples should be different, and following the constraints in $S_{\text{positive}}$, the semantics of the augmentations of different examples should also be different. Therefore, the three terms in $S_{\text{negative}}$ denote that, given an example within a mini-batch, we treat both the other examples and the augmentations derived from other examples as negative examples.

The final loss of SIWCon regularization is computed across all examples in a mini-batch. When SIWCon is used in the normal training scenario (1), the overall objective is:

$$\min_{\theta} \mathcal{L}_{\text{ce}}(X, Y) + \alpha \mathcal{L}_{\text{SIWCon}}(X),$$

where $\alpha$ is a parameter balancing the supervised part and the contrastive regularization part.
4 Experiment

4.1 Metrics

We measure the model performance with Accuracy (ACC.), the model robustness with Accuracy Under Attack (AUA.), and the influence of words with three Area Over the Perturbation Curve (AOPC) metrics (DeYoung et al., 2020; Samek et al., 2017; Nguyen, 2018). AOPC_{\text{Comp.}} and AOPC_{\text{Suff.}} respectively measure the overall influence of the information in important and unimportant words on prediction. AOPC_{\text{Comp.}} is formulated as:

\[
\frac{1}{K+1} \sum_{k=1}^{K} P(Y_{\text{true}}|X) - P(Y_{\text{true}}|t^{\text{imp}}(X)), \quad (9)
\]

and AOPC_{\text{Suff.}} is formulated as:

\[
\frac{1}{K+1} \sum_{k=1}^{K} P(Y_{\text{true}}|X) - P(Y_{\text{true}}|t^{\text{imp}}_{\frac{k}{k}}(X)), \quad (10)
\]

where \(t^{\text{imp}}_{\frac{k}{k}}\) and \(t^{\text{imp}}_k\) are deterministic transformations that remove the \(k\) most and least important words in a sentence, respectively. We also use AOPC_{\text{Diff.}} to indicate the difference between AOPC_{\text{Comp.}} and AOPC_{\text{Suff.}}, measuring how the goal of SIWCon is achieved.

4.2 Experiment Setup

Setup. We conduct experiments on MR (Pang and Lee, 2005), SST2 (Socher et al., 2013), and IMDB (Maas et al., 2011) datasets. We use LSTM (Hochreiter and Schmidhuber, 1997), TextCNN (Kim, 2014), and the base version of BERT (Devlin et al., 2019) as models. More details of the datasets and models can be found in Appendix A.1 and A.2. We use Normal training (1) and Adversarial training (AT, detailed in Appendix A.3) as basic training methods. In the main experiment, we use DeepWordBug (Gao et al., 2018) and TextFooler (Jin et al., 2020) as attack methods. We also use BAE (Garg and Ramakrishnan, 2020), TextBugger (Li et al., 2019), and PWWS (Ren et al., 2019) in the analysis.

Implementation Details. The \(K\) in (9) and (10) are set as 40% of each sentence’s length. We use Adam (Kingma and Ba, 2015) as the optimizer. For LSTM and TextCNN, we use the average token embedding before the last dense layer as the sentence representation. For BERT, we use the [CLS] token embedding as the sentence representation. Unless otherwise specified, the batch size is set as 32, the learning rate/\(\alpha/\tau\) for LSTM, TextCNN, and BERT is 1e-3/1.2/0.01, 1e-3/1.2/0.05, and 3e-5/0.005/1.5. The reported results are the average of five individual runs with randomly picked seeds.

4.3 Main Results

In the main experiment, we train the models on three datasets with different training methods and then measure their robustness by attacking 600 examples randomly picked from the testing set. Following Jin et al. (2020) and Li et al. (2021), for adversarial training, we incorporate the adversarial examples of 10% randomly picked training data into the new training set, which are generated by the same attack method for measuring robustness.
Table 2: The comparisons on the overall influence of the information in the words of different importance on prediction. The bold values of AOPC_Diff indicate the most similar influence and the best achievement of the goal of SIWCon. A is short for AOPC.

SIWCon has a slight impact on clean accuracy. The results of clean accuracy are illustrated in Table 1. SIWCon only slightly impacts the clean accuracy when combined with other training methods. Normal+SIWCon sometimes outperforms Normal method, and the average accuracy difference between the two methods is only 0.97%. AT+SIWCon causes a slight drop in model accuracy compared to Normal method, with the negative impact mainly resulting from the integration of adversarial examples rather than the usage of SIWCon. The average accuracy difference between AT and AT+SIWCon is only 0.36%, and 1.53% between Normal and AT.

SIWCon improves model robustness. The results of robustness are illustrated in Table 1. SIWCon is a self-supervised regularization method that relies solely on the training data (not including labels) and their augmentations generated by removing words, without learning from any adversarial perturbations. Nevertheless, SIWCon is effective in improving model robustness. Under the unforeseen scenario, the average AUA of Normal+SIWCon is 10.60% higher than Normal method (17.45% vs. 6.85%). Under the foreseen scenario, SIWCon can further improve the robustness of models, with the average AUA of AT+SIWCon being 7.35% higher than that of AT (40.98% vs. 33.63%). These results demonstrate the effectiveness of SIWCon and its potential to be combined with more training methods, using as a plug-and-play regularization.

SIWCon makes words of different importance have a similar influence. The results of word influence are illustrated in Table 2. SIWCon similarizes the influence of information in words of different importance, as evidenced by the average AOPC_Diff of Normal+SIWCon being 0.017 lower than that of Normal, and of AT+SIWCon being 0.016 lower than that of AT. Recall the question we raised in section §1, the increased AUA, and decreased AOPC_Diff when using SIWCon in training empirically give an affirmative answer.

4.4 Further Analysis on SIWCon

In this section, we conduct further analysis and ablation study on BERT and MR dataset.

Hyperparameter $\alpha$. The influence of $\alpha$ is illustrated in Figure 3(a). We find that when $\alpha$ is set to different values, the robustness of BERT can always be effectively improved, as the AUA. of Normal+SIWCon is always higher than that of Normal. When $\alpha$ is small, BERT tends to be more robust. Different values of $\alpha$ also have a slight impact on the clean accuracy, as the ACC. of Normal+SIWCon is always close to that of Normal.

Temperature $\tau$. The influence of $\tau$ is shown in Figure 3(b). Similar to $\alpha$, when $\tau$ is set to various values, the robustness of the model is consistently improved, while the ACC. fluctuates around that of the normally trained BERT. However, $\tau$ has a greater impact on the clean accuracy than $\alpha$.

Batch Size. The influence of batch size is shown in Figure 3(c). SIWCon is benefit from larger batch size. As the batch size increases, the gap in clean accuracy (ACC.) between the models trained with DeepWordBug TextFooler

<table>
<thead>
<tr>
<th>Model</th>
<th>Method</th>
<th>DeepWordBug</th>
<th>TextFooler</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$A_{\text{acc}}$</td>
<td>$A_{\text{diff}}$</td>
</tr>
<tr>
<td>MR</td>
<td>Normal</td>
<td>0.096</td>
<td>0.096</td>
</tr>
<tr>
<td></td>
<td>+SIWCon</td>
<td>0.070</td>
<td>0.070</td>
</tr>
<tr>
<td>LSTM</td>
<td>AT</td>
<td>0.084</td>
<td>0.084</td>
</tr>
<tr>
<td></td>
<td>+SIWCon</td>
<td>0.066</td>
<td>0.066</td>
</tr>
<tr>
<td>TextCNN</td>
<td>Normal</td>
<td>0.094</td>
<td>0.094</td>
</tr>
<tr>
<td></td>
<td>+SIWCon</td>
<td>0.084</td>
<td>0.084</td>
</tr>
<tr>
<td>BERT</td>
<td>Normal</td>
<td>0.114</td>
<td>0.114</td>
</tr>
<tr>
<td></td>
<td>+SIWCon</td>
<td>0.114</td>
<td>0.114</td>
</tr>
<tr>
<td>SST2</td>
<td>Normal</td>
<td>0.095</td>
<td>0.095</td>
</tr>
<tr>
<td></td>
<td>+SIWCon</td>
<td>0.095</td>
<td>0.095</td>
</tr>
<tr>
<td>IMDB</td>
<td>Normal</td>
<td>0.090</td>
<td>0.090</td>
</tr>
<tr>
<td></td>
<td>+SIWCon</td>
<td>0.090</td>
<td>0.090</td>
</tr>
</tbody>
</table>

7896
Figure 3: Influence of the hyperparameter \( \alpha \), temperature \( \tau \), and batch size.

Figure 4: The comparisons of ACC. and AUA. under different adversarial training setting. 0\% denotes no adversarial examples are generated, and AT downgrades to Normal, which is the unforeseen scenario. Other ratios indicate the number of adversarial examples incorporated in training, which is the foreseen scenario.

and without SIWCon decreases, while the gap in robustness (AUA.) tends to increase. We conjecture that this is due to the contrastive nature of SIWCon regularization, as larger batch sizes provide more negative examples, thereby facilitating the regularizing (Chen et al., 2020a).

**Attack Methods and Examples Ratio.** We test the performance of SIWCon with more attack methods under different adversarial training settings, and the results are shown in Figure 4. We observe that SIWCon consistently outperforms the basic training method in terms of model robustness when using different attack methods. Additionally, we find that when a higher proportion of adversarial examples are incorporated into adversarial training, robustness may sometimes be reduced. However, SIWCon effectively mitigates this negative impact.

**Ablation Study.** We replace the data augmentation operations \( \hat{t}^{imp}(\cdot) \) and \( \hat{t}^{ump}(\cdot) \) in SIWCon with new augmentation operations that randomly drop out words in sentences to perform ablation study. The results in Table 3 show that the influence-based data augmentation operations used in SIWCon help the model (i) improve robustness, as AUA. of SIWCon are higher than the random methods, and (ii) similarize the influence of the words of different importance on prediction, as AOPC\text{Diff.} of SIWCon are lower than the random methods.

**4.5 Further Analysis on Model Behavior**

**Loss Landscape.** Following the filter normalization scheme proposed by Li et al. (2018), we fine-tune BERT on the MR training set, and plot the loss landscape of BERT on the MR testing set, as shown in Figure 5. It is shown that the loss landscape of Normal+SIWCon (b) is visibly smoother and changes more slowly than the normally trained BERT (a). Furthermore, adversarial training (c) makes the loss landscape smoother than the Normal method (a), while when it is combined with SIWCon (d), the loss landscape is further smoothened. According to the finding of Mok et al. (2021) that a robust model should have a smooth loss landscape, the visualization results demonstrate that SIWCon is effective for improving model robustness.

**Sentence Representation.** We fine-tune BERT on MR and then, for a normal sentence, we generate two groups of sentences by cumulatively removing the 40% most and least important words in the sentence (e.g., abcd \( \rightarrow \) abcd \( \rightarrow \) abcd), following the gradient attribution (3). We also utilize PWWS (Ren et al., 2019) to generate adversarial examples from the normal sentence. The sentence representations visualized by t-SNE (van der Maaten and Hinton, 2008) and the reduction paths (Feng et al., 2018) are shown in Figure 6.
Figure 5: The loss landscape of BERT tuned with different methods.

Figure 6: The visualization of sentence representations and reduction paths. The results are obtained from the MR instance “A sports movie with action that’s exciting on the field and a story you care about off it.” Darker examples indicate more words are removed. Black and orange arrows respectively illustrate the reduction path of unimportant and important words. Blue arrow highlights the reduction that drastically biases the prediction. Representations in pink area belong to the neighborhood of the adversarial example.

More results can be found in Appendix B.1.

The representation of the normal sentence (●) can be seen as a point with complete information for supporting the prediction contained, the bias of incomplete sentences (∆ and □) from the normal sentence (●) can be seen as the information loss caused by word removal, and the location of the adversarial example indicates when how much information is lost, the example can no longer maintain the original prediction. When unimportant words are removed (□), the representations for both models are steadily biased from the normal sentence, and removal will not drastically bias the representations towards the adversarial example, indicating that the information in unimportant words is not influential on prediction. However, the two models behave differently when important words are removed (∆). For Normal method, the representations are biased towards the adversarial example, and the prediction will be drastically biased when a few important words are removed (indicated by the blue arrow). For SIWCon, the representations are steadily biased in a similar manner as when unimportant words are removed, and the representations do not fall into the neighborhood of the adversarial example, indicating that important words are less influential on prediction and it is more difficult for attack methods to find adversarial examples.

Confidence Changing. We illustrate the change in model confidence with the removal of words on case instance in Figure 7. More results can be found in Appendix B.2. We cumulatively remove the most or least important words in a sentence, and the change in confidence can be seen as the influence of the information in the removed words. SIWCon reduces the influence of the information in important words, as more important words need to be removed to shift the model’s prediction.

5 Conclusion

This paper presents SIWCon, a self-supervised regularization method based on contrastive learning. SIWCon improves the robustness of language models by encouraging the words of different importance to have more similar influence on prediction. Experiments show that SIWCon effectively improves model robustness without depending on adversarial perturbation. We hope the insights provided in this paper will inspire further research.
Limitations

The loss objective of the proposed SIWCon regularization is computed on augmented data, which increases the time required for the model to complete training. We evaluate SIWCon on classification tasks, but it may be applied to various other tasks, such as reading comprehension and textual entailment. More evaluations are expected to be done in future works. The proposed SIWCon regularization is effective in defending against word-level adversarial attacks, as the basic elements of the augmentation methods are words. However, similar regularization techniques can also be applied to characters and sentences, and we leave evaluating the effectiveness of such variants in future works.

Ethics Statement

In this paper, we propose a self-supervised regularization method for improving the model robustness, which does not need to learn from any adversarial examples. Since adversarial examples are always difficult to generate for language models, our method can thus reduce the financial and environmental cost of robustness improvement. Furthermore, our method forces models consider different words to have a similar degree of influence on prediction, potentially reducing the model’s bias. All the datasets we use are publicly available, and we do not violate their licenses.

Acknowledgements

The authors would like to thank the anonymous reviewers for their comprehensive and constructive comments. This research was supported by National Research and Development Program of China (No.2019YFB1005200).

References


A Additional Experimental Details

A.1 Details on Dataset
MR contains movie reviews from Rotten Tomatoes, and the examples are labeled as positive or negative, with 8,530 for training and 1,066 for testing. SST2 contains sentences labeled as positive or negative, with 67,349 for training and 1,821 for testing. IMDB contains binary polar movie reviews from Internet Movie Database, which are also labeled as positive or negative, with 25,000 for training and 25,000 for testing.

A.2 Details on Model
The experiments are conducted on three models with different architectures. The LSTM (Hochreiter and Schmidhuber, 1997) consists of a 300-dimensional GloVe embedding layer (Pennington et al., 2014), a Bi-LSTM layer with 150 hidden units, and a dense layer. The TextCNN is similar to the architecture in (Kim, 2014), while the embedding is also replaced with the 300-dimensional GloVe embedding. The BERT (Devlin et al., 2019) used in our experiment is the base uncased version.

A.3 Details on Baseline
When SIWCon is combined with adversarial training, the overall objective is formulated as:

$$\min_{\theta} \mathcal{L}_{ce}(X, Y) + \mathcal{L}_{adv}(X^{adv}, Y) + \alpha \mathcal{L}_{SIWCon}(X). \quad (11)$$

This joint training objective helps the model to learn both the normal and adversarial examples distribution and simultaneously regularizes the model on the word influence.

B Additional Experimental Results

B.1 Analysis on Sentence Representation
We give more visualizations of sentence representations and reduction paths in Figure 8-13. The instance sentences are randomly picked from MR dataset, and the results are obtained on BERT.

B.2 Analysis on Confidence Changing
We provide more results on the change in model confidence with the removal of words in 14-17. The instance sentences are randomly picked from MR dataset, and the results are obtained on BERT.

Figure 8: The visualization of sentence representations and reduction paths. The results are obtained on the BERT sentences representation of the MR instance “I enjoyed time of favor while I was watching it, but I was surprised at how quickly it faded from my memory.”

Figure 9: The visualization of sentence representations and reduction paths. The results are obtained on the BERT sentences representation of the MR instance “If nothing else, this movie introduces a promising, unusual kind of psychological horror.”
Figure 10: The visualization of sentence representations and reduction paths. The results are obtained on the BERT sentences representation of the MR instance “Everytime you think undercover brother has run out of steam, it finds a new way to surprise and amuse.”

Figure 11: The visualization of sentence representations and reduction paths. The results are obtained on the BERT sentences representation of the MR instance “A real movie, about real people, that gives us a rare glimpse into a culture most of us don’t know.”

Figure 12: The visualization of sentence representations and reduction paths. The results are obtained on the BERT sentences representation of the MR instance “There’s a lot of tooth in roger dodger. but what’s nice is that there’s a casual intelligence that permeates the script.”

Figure 13: The visualization of sentence representations and reduction paths. The results are obtained on the BERT sentences representation of the MR instance “This is the best American movie about troubled teens since 1998’s whatever.”

Figure 14: The change in model confidence with the removal of words until label shift. The results are obtained on the MR instance “The entire movie has a truncated feeling, but what’s available is lovely and lovable.”

Figure 15: The change in model confidence with the removal of words until label shift. The results are obtained on the MR instance “I enjoyed time of favor while i was watching it, but I was surprised at how quickly it faded from my memory.”

Figure 16: The change in model confidence with the removal of words until label shift. The results are obtained on the MR instance “Some actors have so much charisma that you’d be happy to listen to them reading the phone book. Hugh grant and Sandra bullock are two such likeable actors.”

Figure 17: The change in model confidence with the removal of words until label shift. The results are obtained on the MR instance “An engaging overview of Johnson’s eccentric career.”
ACL 2023 Responsible NLP Checklist

A For every submission:

✓ A1. Did you describe the limitations of your work?
   In Section Limitations.

✓ A2. Did you discuss any potential risks of your work?
   In Section Ethics Statement.

✓ A3. Do the abstract and introduction summarize the paper’s main claims?
   In 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?
   In Section 3 and Section 4.

✓ B1. Did you cite the creators of artifacts you used?
   In Section 1, Section 4.2, and Appendix A.

✓ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
   In Section Ethics Statement.

✓ 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)?
   In Section Ethics Statement.

✗ 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?
   The datasets we used in the paper are widely used benchmark datasets.

✓ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
   In Section 4.2, Appendix A.1, and Appendix A.2.

✓ 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.
   In Appendix A.1

C ✓ Did you run computational experiments?
   In Section 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?
   In Section Limitations and Appendix A.2.

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?
   In Section 4.2, Appendix A.

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?
   In Section 4.2.

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.)?
   In Section 4.2.

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.