RobustSentEmbed: Robust Sentence Embeddings Using Adversarial Self-Supervised Contrastive Learning

Javad Rafiei Asl,[1] Prajwal Panzade,[1] Eduardo Blanco,[2] Daniel Takabi,[3] Zhipeng Cai[1]
[1]Georgia State University, [2]University of Arizona, [3]Old Dominion University
jasl1@student.gsu.edu, ppanzade1@student.gsu.edu, eduardoblanco@arizona.edu,
takabi@odu.edu, zcai@gsu.edu

Abstract

Pre-trained language models (PLMs) have consistently demonstrated outstanding performance across a diverse spectrum of natural language processing tasks. Nevertheless, despite their success with unseen data, current PLM-based representations often exhibit poor robustness in adversarial settings. In this paper, we introduce RobustSentEmbed, a self-supervised sentence embedding framework designed to improve both generalization and robustness in diverse text representation tasks and against a diverse set of adversarial attacks. Through the generation of high-risk adversarial perturbations and their utilization in a novel objective function, RobustSentEmbed adeptly learns high-quality and robust sentence embeddings. Our experiments confirm the superiority of RobustSentEmbed over state-of-the-art representations. Specifically, Our framework achieves a significant reduction in the success rate of various adversarial attacks, notably reducing the BERTAttack success rate by almost half (from 75.51% to 38.81%). The framework also yields improvements of 1.59% and 0.23% in semantic textual similarity tasks and various transfer tasks, respectively.

1 Introduction

Pre-trained Language Models (PLMs) have demonstrated state-of-the-art performance in learning contextual word embeddings (Devlin et al., 2019), contributing to significant advancements in various Natural Language Processing (NLP) tasks (Yang et al., 2019; He et al., 2021; Ding et al., 2023). PLMs, including prominent models like BERT (Devlin et al., 2019) and GPT-3 (Brown et al., 2020), have revolutionized text classification, sentence representation, and machine translation among a plethora of diverse NLP tasks. While PLMs have expanded their focus to include universal sentence embeddings, which effectively capture the semantic representation of input text, PLM-based sentence representations lack two crucial characteristics: generalization and robustness.

Extensive research efforts have been dedicated to the development of universal sentence embeddings employing PLMs (Reimers and Gurevych, 2019; Zhang et al., 2020; Neelakantan et al., 2022; Wang et al., 2023). Although these embeddings have demonstrated proficiency in generalization across various downstream tasks (Sun et al., 2019; Gao et al., 2021), they exhibit limitations when subjected to adversarial settings and remain vulnerable to adversarial attacks (Nie et al., 2020; Wang et al., 2021). Existing research has highlighted the limited robustness of PLM-based representations (Garg and Ramakrishnan, 2020; Wu et al., 2023; Hauser et al., 2023). The vulnerability arises when these representations can be easily deceived by making small, imperceptible modifications to the input text.

To address these limitations, we propose a method to obtain robust sentence embeddings called RobustSentEmbed. The main idea is to generate small adversarial perturbations and employ an efficient contrastive objective (Chen et al., 2020). The goal is to enhance the adversarial resilience of the sentence embeddings. Specifically, our framework involves an iterative collaboration between an adversarial perturbation generator and the PLM-based encoder to generate high-risk perturbations in both token-level and sentence-level embedding spaces. RobustSentEmbed then employs a contrastive learning objective in conjunction with a token replacement detection objective to maximize the similarity between the embedding of the original sentence and the adversarial embedding of a positive pair (the former objective) as well as its edited sentence (the latter objective).

We have conducted comprehensive experiments to substantiate the efficacy of the RobustSentEmbed framework. The tasks encompass TextAttack (Morris et al., 2020) assessments, adversar-

Findings of the Association for Computational Linguistics: NAACL 2024, pages 3795–3809

ial Semantic Textual Similarity (STS) tasks, Non-adversarial STS tasks (Conneau and Kiela, 2018), and transfer tasks (Conneau and Kiela, 2018). Two initial series of experiments were designed to evaluate the robustness of our sentence embeddings against various adversarial attacks and tasks. Subsequently, we conducted two final series of experiments to assess the quality of our embeddings in the contexts of semantic similarity and natural language understanding. RobustSentEmbed demonstrates significant improvements in robustness, reducing the attack success rate from 75.51% to 38.81% against the BERTAttack attack and from 71.86% to 12.80% on adversarial STS. Moreover, the framework outperforms existing methods in ten out of thirteen tasks while obtaining comparable results with the other three, showcasing improvements of 1.59% and 0.23% on STS tasks and NLP transfer tasks, respectively.

Contributions. Our main contributions are summarized as follows:

- We introduce RobustSentEmbed, an innovative framework designed for generating sentence embeddings that are robust against adversarial attacks. Existing methods are vulnerable to such adversarial challenges. RobustSentEmbed fills this gap by generating high-risk perturbations and utilizing an efficient adversarial objective function.[1]

- We conduct comprehensive experiments to empirically evaluate the effectiveness of the RobustSentEmbed framework. The empirical findings substantiate the efficacy of our framework, as demonstrated by its superior performance in both robustness and generalization benchmarks.

2 Related Work

Recently, self-supervised methods using contrastive objectives have become prominent for learning effective and robust text representations: SimCSE, as outlined by Gao et al. (2021), introduced a minimal augmentation method involving the application of two distinct dropout masks to predict the input sentence. The ConSERT model (Yan et al., 2021) employed four unique data augmentation techniques, namely adversarial attacks, token shuffling, cut-off, and dropout, to generate a variety of perspectives in order to carry out a contrastive objective. Miao et al. (2021) utilized adversarial training to improve the robustness of contrastive learning. They achieved this by incorporating regularization into their learning objective, combining benign contrastive learning with an adversarial contrastive scenario. Rima et al. (2022) proposed a novel method for training language processing models, combining adversarial training and contrastive learning. Their approach incorporates linear perturbations to input embeddings and uses contrastive learning to minimize the distance between the original and perturbed representations. Pan et al. (2022) introduced a simple technique to improve the fine-tuning of Transformer-based encoders. Their method involves regularization by generating adversarial examples through word embedding perturbations and using contrastive learning to obtain noise-invariant representations.

Unlike existing approaches for training text representation through contrastive adversarial learning (Yan et al., 2021; Miao et al., 2021; Rima et al., 2022; Pan et al., 2022), our framework generates more efficient, high-risk perturbations at both the token-level and sentence-level within the embedding space. Furthermore, our framework utilizes a robust contrastive objective and incorporates an adversarial replaced token detection method, leading to high-quality text representations that yield improved generalization and robustness characteristics.

3 The Proposed Framework

We extended our previous robust text representation (i.e., RobustEmbed, see (Asl et al., 2023)) by utilizing adversarial perturbation at various levels, including token-level and sentence-level. We introduce RobustSentEmbed, a straightforward yet highly effective method for generating robust text representation. Given a PLM $f_\theta(.)$ as the encoder and a raw dataset $\mathcal{D}$, our framework aims to pre-train $f_\theta(\cdot)$ on $\mathcal{D}$ to enhance the efficacy of sentence embeddings across a wide range of NLP tasks (improved generalization) and to fortify its resilience against various adversarial attacks (improved robustness). Figure 1 presents an overview of our framework. The framework involves an iterative interaction between the perturbation generator and the $f_\theta(.)$ encoder to produce high-risk adversarial perturbations in both token-level and sentence-level embedding spaces. These perturbations provide the

[1] Our code are publicly available at `https://github.com/jasl1/RobustSentEmbed`

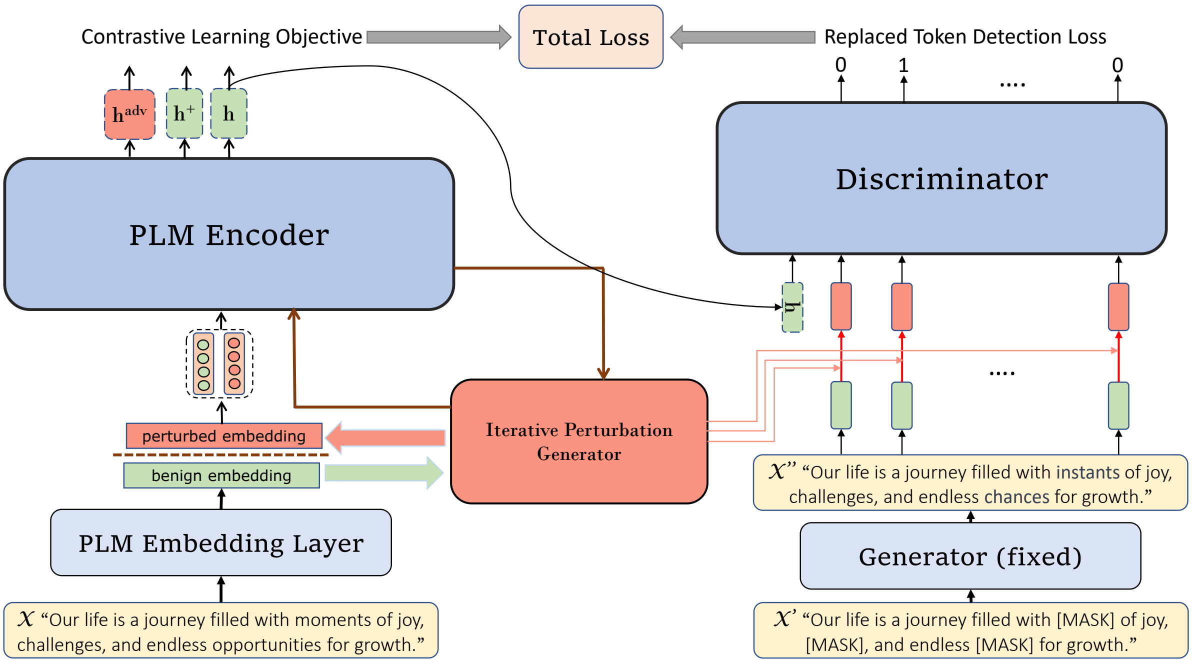

Figure 1: The general architecture of the RobustSentEmbed framework.

essential adversarial examples required for adversarial training by both the $f_\theta(.)$ encoder and a PLM-based discriminator. The subsequent sections will delve into the main components of our framework.

3.1 Perturbation Generator

Adversarial perturbation involves adding maliciously crafted perturbations into benign data, with the objective of misleading Machine Learning (ML) models (Goodfellow et al., 2015). A highly effective and broadly applicable method for generating adversarial perturbations is to apply a small noise $\boldsymbol{\delta}$ within a norm-constraint ball, aiming to maximize the adversarial loss function:

$$\arg\max_{||\boldsymbol{\delta}||\leq\epsilon} L(f_\theta(X+\boldsymbol{\delta}), y), \tag{1}$$

where $f_\theta(.)$ denotes an ML model parameterized with X as the sub-word embeddings. There are numerous gradient-based algorithms designed to address this optimization problem. Our framework extends the token-level perturbation method proposed by Li and Qiu (2021) by complementing the perturbation with an innovative sentence-level perturbation generator in order to generate worst-case adversarial examples. The main idea is to train a PLM-based model to withstand a broad spectrum of adversarial attacks, spanning both word and instance levels.

Recognizing the different roles that individual tokens play within a sentence, the RobustSentEmbed framework incorporates a scaling index to allow larger perturbations for tokens exhibiting larger gradients during the normalization of token-level perturbations:

$$n^i = \frac{\|\boldsymbol{\eta}_i^t\|_P}{\max_j \|\boldsymbol{\eta}_j^t\|_P}, \tag{2}$$

where $\boldsymbol{\eta}_i^t$ represents the token-level perturbation for word i at step t of the gradient ascent, and P denotes the type of norm constraint. Considering the encoder $f_\theta(.)$ and an input sentence x, RobustSentEmbed passes the sentence through $f_\theta(.)$ by applying standard dropout twice. This process yields two different embeddings, denoted as "positive pairs" and represented as (X, X^+). Finally, the newly adjusted token-level perturbation is formulated as:

$$\boldsymbol{\eta}_i^{t+1} = n^i * (\boldsymbol{\eta}_i^t + \gamma \frac{\boldsymbol{g}_{\eta_i}}{\|\boldsymbol{g}_{\eta_i}\|_P)}, \tag{3}$$

$$\boldsymbol{\eta}^{t+1} \leftarrow \Pi_{\|\boldsymbol{\eta}\|_P \leq \epsilon}(\boldsymbol{\eta}^t), \tag{4}$$

where $\boldsymbol{g}_{\eta_i} = \nabla_\eta \mathcal{L}_{con,\theta}(\boldsymbol{X} + \boldsymbol{\delta}^{t-1} + \boldsymbol{\eta}^{t-1}, \{\boldsymbol{X}^+\})$ is the gradient of the contrastive learning loss with respect to $\boldsymbol{\eta}$. The perturbation is generated by the ℓ_∞ norm-ball with radius ϵ, and Π projects the perturbation onto the ϵ-ball.

To generate adversarial perturbations at the sentence-level, RobustSentEmbed employs a combination of the Fast Gradient Sign Method (FGSM)

(Goodfellow et al., 2015) and the Projected Gradient Descent (PGD) technique (Madry et al., 2018). The framework iterates using this combination, specifically T-step FGSM and K-step PGD, to systematically reinforce invariance within the embedding space. Ultimately, this strategy leads to enhanced generalization and robustness. It proceeds with the following steps to update the perturbation for PGD in iteration $k+1$ and FGSM in iteration $t+1$:

$$\boldsymbol{\delta}^{k+1}_{\text{pgd}} = \Pi_{\|\boldsymbol{\delta}\|_P \leq \epsilon}(\boldsymbol{\delta}^k + \alpha g(\boldsymbol{\delta}^k)/\|g(\boldsymbol{\delta}^k)\|_P), \quad (5)$$

$$\boldsymbol{\delta}^{t+1}_{\text{fgsm}} = \Pi_{\|\boldsymbol{\delta}\|_P \leq \epsilon}(\boldsymbol{\delta}^t + \beta \text{sign}(g(\boldsymbol{\delta}^t))), \quad (6)$$

where $g(\boldsymbol{\delta}^n) = \nabla_{\boldsymbol{\delta}} \mathcal{L}_{con,\theta}(\boldsymbol{X} + \boldsymbol{\delta}^n, \{\boldsymbol{X}^+\})$ with $n = t \ or \ k$ represents the gradient of the contrastive learning loss with respect to $\boldsymbol{\delta}$. The variables α and β denote the step sizes for the attacks, while $\text{sign}(.)$ yields the vector's sign. The final perturbation is obtained by employing a practical combination of T-step FGSM and K-step PGD:

$$\boldsymbol{\delta}_{\text{final}} = \rho \boldsymbol{\delta}^K_{\text{pgd}} + (1-\rho)\boldsymbol{\delta}^T_{\text{fgsm}}, \quad (7)$$

where $0 \leq \rho \leq 1$ modulates the relative importance of each separate perturbation in the formation of the final perturbation.

3.2 Robust Contrastive Learning

To achieve robust text representations through adversarial learning, we employ a straightforward approach that can be described as the combination of a Replaced Token Detection (RTD) objective (Figure 1, right) with a novel self-supervised contrastive learning objective (Figure 1, left).

Our framework extends an adversarial version of the RTD task used in ELECTRA (Clark et al., 2020). In this approach, given an input sentence x, ELECTRA utilizes a pre-trained masked language model as the generator G to recover randomly masked tokens in $x^{'} = \text{Mask}(x)$, resulting in the edited sentence $x^{''} = G(x^{'})$. Subsequently, a discriminator D is tasked with predicting whether token replacements have occurred, which constitutes the RTD task. As illustrated in Figure 1, the perturbation generator module introduces token-aware perturbations into the embedding of each individual token, making it more challenging for discriminator D to perform the RTD task effectively. The gradient of D can be back-propagated into f through $\boldsymbol{h} = f_\theta(x)$. This mechanism encourages f to make vector $\boldsymbol{h}$ sufficiently informative, enhancing its resilience against token-level adversarial attacks. Consequently, our framework employs the following adversarial objective for a single sentence x:

$$\begin{aligned}\mathcal{L}^x_{RTD} = \sum_{j=1}^{|x|} [&-\mathbb{1}(X^{adv}_j = X_j) \log D(X^{adv}, \boldsymbol{h}, j) \\ &-\mathbb{1}(X^{adv}_j \neq X_j) \log (1 - D(X^{adv}, \boldsymbol{h}, j))], \quad (8)\end{aligned}$$

where $X^{adv} = X'' + \boldsymbol{\eta}_i^{max(K,T)}$ represent the ith perturbed token in x. The training objective for the batch B is $\mathcal{L}_{RTD,\theta} = \sum_{i=1}^{|B|} \mathcal{L}^{x_i}_{RTD}$. Furthermore, we use self-supervised contrastive learning to acquire effective low-dimensional representations by bringing semantically similar samples closer and pushing dissimilar ones further apart. Let $\{(x_i, x_i^+)\}_{i=1}^N$ denote a set of N positive pairs, where x_i and x_i^+ are semantically correlated and (z_i, z_i^+) represents the corresponding embedding vectors for the positive pair (x_i, x_i^+). We define z_i's positive set as $z_i^{pos} = \{z_i^+\}$, while the negative set $z_i^{neg} = \{z_i^-\}$ is the set of positive pairs from other sentences in the same batch. Then, the contrastive training objective is defined as follows:

$$\begin{aligned}&\mathcal{L}_{con,\theta}(z_i, z_i^{pos}, z_i^{neg}) = \\ &-\log(\frac{\sum_{z_i^{pos}} \exp(sim(z_i, z_i^+)/\tau)}{\sum_{(z_i^{pos} \cup z_i^{neg})} \exp(sim(z_i, z_i^{+\ or\ -})/\tau)}), \quad (9)\end{aligned}$$

where τ denotes a temperature hyperparameter and $sim(u, v) = \frac{u^\top v}{\|u\| \cdot \|v\|}$ is the cosine similarity between two representations. Our framework utilizes contrastive learning to maximize the similarity between clean examples and their adversarial perturbation by incorporating the adversarial example as an additional element within the positive set:

$$\mathcal{L}_{RobustSentEmbed,\,\theta} := \mathcal{L}_{con,\theta}(z, \{z^{pos},\ z^{adv}\}, \{z^{neg}\}).$$

$$\begin{aligned}\mathcal{L}_{total} :=& \mathcal{L}_{RobustSentEmbed,\theta} + \lambda_1 \cdot \mathcal{L}_{con,\theta}(z^{adv}, \{z^{pos}\}, \{z^{neg}\}) \\ &+ \lambda_2 \cdot \mathcal{L}_{RTD,\theta},\end{aligned} \quad (10)$$

where $z^{adv} = z + \boldsymbol{\delta}_{final}$ represents the adversarial perturbation of the input sample x in the embedding space, and λ_1, λ_2 denote weighting coefficients. The first component of the total contrastive loss (Eq. 10) is designed to optimize the sentence-level similarity between the input sample x, its positive pair, and its adversarial perturbation, while the second component serves to regularize the loss by encouraging the convergence of the adversarial perturbation and the positive pair of x. The final component introduces the adversarial Replaced Token Detection (RTD) objective into the total contrastive loss.

Adversarial Attack	Model	IMDB	MR	SST2	YELP	MRPC	SNLI	MNLI-Mismatched	Avg.
TextFooler	SimCSE-BERT$_{\text{base}}$	75.32	65.53	71.49	79.67	80.07	72.65	68.54	72.61
	USCAL-BERT$_{\text{base}}$	61.94	48.71	55.38	62.30	60.18	54.82	53.74	56.72
	RobustEmbed-BERT$_{\text{base}}$	40.55	32.69	36.17	44.25	38.88	37.61	35.63	37.97
	RobustSentEmbed-BERT$_{\text{base}}$	**40.02**	**31.39**	**35.83**	**43.78**	**37.54**	**36.99**	**34.15**	**37.10**
TextBugger	SimCSE-BERT$_{\text{base}}$	52.21	42.04	49.67	56.19	56.73	45.39	40.16	48.91
	USCAL-BERT$_{\text{base}}$	39.16	27.37	31.90	41.25	37.86	30.79	25.45	33.40
	RobustEmbed-BERT$_{\text{base}}$	23.70	18.03	20.24	28.58	20.89	19.07	16.33	20.98
	RobustSentEmbed-BERT$_{\text{base}}$	**23.16**	**17.49**	**19.62**	**27.93**	**19.37**	**18.05**	**15.51**	**20.16**
PWWS	SimCSE-BERT$_{\text{base}}$	64.41	55.73	60.48	67.54	68.15	56.09	52.58	60.71
	USCAL-BERT$_{\text{base}}$	51.95	40.67	45.29	52.30	46.86	50.92	39.37	46.77
	RobustEmbed-BERT$_{\text{base}}$	33.63	28.15	30.56	29.94	25.51	27.16	28.49	29.06
	RobustSentEmbed-BERT$_{\text{base}}$	**32.94**	**28.05**	**29.28**	**29.14**	**24.72**	**26.28**	**27.90**	**28.33**
BAE	SimCSE-BERT$_{\text{base}}$	73.50	61.83	68.27	75.15	77.84	69.06	65.43	70.15
	USCAL-BERT$_{\text{base}}$	58.57	46.19	51.72	59.49	58.38	50.90	51.16	53.77
	RobustEmbed-BERT$_{\text{base}}$	37.35	29.82	32.08	41.66	36.45	34.17	31.98	34.79
	RobustSentEmbed-BERT$_{\text{base}}$	**37.16**	**29.12**	**31.43**	**40.96**	**35.53**	**33.87**	**31.85**	**34.27**
BERTAttack	SimCSE-BERT$_{\text{base}}$	78.42	66.94	73.59	80.87	82.16	74.35	72.22	75.51
	USCAL-BERT$_{\text{base}}$	63.23	51.08	57.73	63.96	63.05	55.41	55.86	58.62
	RobustEmbed-BERT$_{\text{base}}$	42.30	34.76	38.81	45.15	39.97	39.08	37.24	39.62
	RobustSentEmbed-BERT$_{\text{base}}$	**41.51**	**34.19**	**38.16**	**44.96**	**38.26**	**38.60**	**35.98**	**38.81**

Table 1: Attack success rates (lower is better) of various adversarial attacks applied to four sentence embeddings (SimCSE, USCAL, RobustEmbed, and RobustSentEmbed) across five text classification and two natural language inference tasks. RobustSentEmbed reduces the attack success rate to less than half across all attacks.

4 Evaluation and Experimental Results

This section presents a comprehensive set of experiments conducted to validate the proposed framework's effectiveness in terms of robustness and generalization metrics. To evaluate robustness, the experiments include adversarial attacks and adversarial Semantic Textual Similarity (STS) tasks. To evaluate generalization, the experiments include non-adversarial STS and transfer tasks within the SentEval framework.[2] Appendices A and B provide training details and ablation studies that illustrate the effects of hyperparameter tuning.

4.1 Adversarial Attacks

We evaluate the robustness of our framework against various adversarial attacks, comparing it with two state-of-the-art sentence embedding models: SimSCE (Gao et al., 2021) and USCAL (Miao et al., 2021). We fine-tuned the BERT-based PLM across seven text classification and natural language inference tasks, specifically MRPC (Dolan and Brockett, 2005), YELP (Zhang et al., 2015), IMDb (Maas et al., 2011), Movie Reviews (MR) (Pang and Lee, 2005), SST2 (Socher et al., 2013), Stanford NLI (SNLI) (Bowman et al., 2015), and Multi-NLI (MNLI) (Williams et al., 2018). To assess the robustness of our fine-tuned model, we investigated the impact of five popular adversarial attacks: TextBugger (Li et al., 2019), PWWS (Ren et al., 2019), TextFooler (Jin et al., 2020), BAE (Garg and Ramakrishnan, 2020), and BERTAttack (Li et al., 2020b). Additional information of these attacks is provided in Appendix C. To ensure statistical validity, we conducted each experiment five times, with each iteration comprising 1000 adversarial attack samples.

Table 1 presents the average attack success rates of five adversarial attacks applied to four sentence embeddings including our previous RobustEmbed method (Asl et al., 2023). Notably, our embedding framework consistently outperforms the other two embedding methods (i.e. SimSCE and USCA), demonstrating significantly lower attack success rates (less than half) across all text classification and natural language inference tasks. Consequently, RobustSentEmbed achieves the lowest average attack success rate against all adversarial attack techniques. Moreover, our framework achieves slightly better performance compared to our previous RobustEmbed framework. These findings substantiate the robustness of our embedding framework and highlight the vulnerabilities of other state-of-the-art sentence embeddings when con-

[2] https://github.com/facebookresearch/SentEval

Adversarial Attack	Model	AdvSTS-B	AdvSICK-R	Avg.
TextFooler	SimCSE-BERT$_{base}$	21.07	24.17	22.62
	USCAL-BERT$_{base}$	16.52	18.71	17.62
	RobustEmbed-BERT$_{base}$	7.48	8.95	8.22
	RobustSentEmbed-BERT$_{base}$	**7.18**	**8.53**	**7.86**
TextBugger	SimCSE-BERT$_{base}$	27.49	28.34	27.91
	USCAL-BERT$_{base}$	21.52	24.88	23.20
	RobustEmbed-BERT$_{base}$	11.76	13.01	12.39
	RobustSentEmbed-BERT$_{base}$	**11.32**	**12.94**	**12.13**
PWWS	SimCSE-BERT$_{base}$	24.15	26.82	25.49
	USCAL-BERT$_{base}$	21.28	23.65	22.47
	RobustEmbed-BERT$_{base}$	13.56	14.44	14.00
	RobustSentEmbed-BERT$_{base}$	**12.68**	**13.90**	**13.29**
BAE	SimCSE-BERT$_{base}$	26.92	28.81	27.86
	USCAL-BERT$_{base}$	22.92	25.48	24.20
	RobustEmbed-BERT$_{base}$	11.13	12.82	11.98
	RobustSentEmbed-BERT$_{base}$	**10.53**	**12.09**	**11.31**
BERTAttack	SimCSE-BERT$_{base}$	31.60	32.85	32.23
	USCAL-BERT$_{base}$	26.02	28.51	27.26
	RobustEmbed-BERT$_{base}$	12.99	13.18	13.09
	RobustSentEmbed-BERT$_{base}$	**12.58**	**13.02**	**12.80**

Table 2: Attack success rates (lower is better) of five adversarial attack techniques applied to four sentence embeddings (SimCSE, USCAL, RobustEmbed, and RobustSentEmbed) across two Adversarial Semantic Textual Similarity (AdvSTS) tasks (i.e. AdvSTS-B and AdvSICK-R). RobustSentEmbed reduces the attack success rate to less than half across all attacks.

fronted with various adversarial attacks.

Figure 2 presents the results of 1000 attacks conducted on two fine-tuned sentence embeddings, assessing the average number of queries required and the resulting accuracy reduction. Attacks on the RobustSentEmbed framework are represented by green data points, while red points denote attacks on the USCAL approach (Miao et al., 2021). Each pair of connected points corresponds to a specific attack. Ideally, a robust sentence embedding should be positioned in the top-left region of the graph, indicating that it necessitates a higher number of queries for an attack to deceive the model while causing minimal performance degradation. Across all adversarial attacks, RobustSentEmbed consistently exhibits greater stability compared to the USCAL method. In other words, a larger number of queries is required for RobustSentEmbed, resulting in a lower accuracy reduction (i.e., better performance) compared to USCAL.

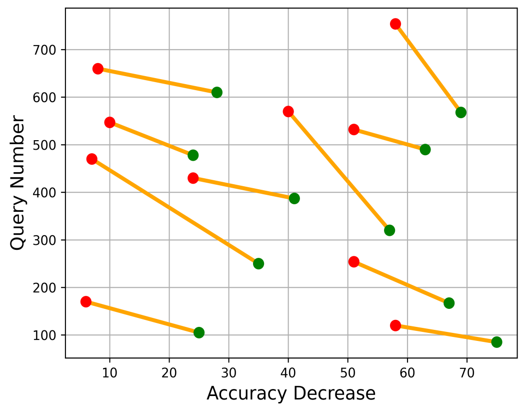

Figure 2: Average number of queries and the resulting accuracy reduction for two fine-tuned embeddings.

4.2 Robust Embeddings

We introduce a new task named Adversarial Semantic Textual Similarity (AdvSTS) to assess the robustness of sentence embeddings. AdvSTS leverages an efficient adversarial technique, like TextFooler, to manipulate an input sentence pair of a non-adversarial STS task in a manner that leads the target model to generate a regression score that maximally deviates from the actual score (truth label). As a result, we generate an adversarial STS dataset by transforming all benign instances from the original (i.e. non-adversarial) dataset into adversarial examples. Table 2 presents the attack success rates of five adversarial attacks applied to four sentence embeddings, including our framework including our previous RobustEmbed method (Asl et al., 2023). These evaluations are conducted for two AdvSTS tasks, specifically AdvSTS-B (originated from STS Benchmark (Cer et al., 2017)) and AdvSICK-R (originated from SICK-Relatedness (Marelli et al., 2014)). Notably, our framework consistently outperforms the other two sentence embedding methods, exhibiting significantly lower attack success rates across both AdvSTS tasks and all employed adversarial attacks. Our framework also demonstrates a slightly enhanced performance in comparison to our earlier RobustEmbed framework. These results provide additional evidence supporting the notion that RobustSentEmbed generates robust text representation.

4.3 Semantic Textual Similarity (STS) Tasks

In this section, we assess the performance of our framework across seven Semantic Textual Similarity (STS) tasks encompassing STS datasets from 2012 to 2016 (Agirre et al., 2012, 2013, 2014, 2015, 2016), STS Benchmark, and SICK-Relatedness. To benchmark our framework's effectiveness, we conducted a comparative analysis against a range of unsupervised sentence embedding approaches, including: 1) baseline methods such as GloVe (Pennington et al., 2014) and average BERT embeddings; 2) post-processing methods like BERT-flow (Li et al., 2020a) and BERT-whitening (Su et al., 2021); and 3) state-of-the-art methods such as SimCSE (Gao et al., 2021), USCAL (Miao et al., 2021), and also our RobustEmbed framework (Asl et al., 2023). We validate the findings of the SimCSE, ConSERT, and USCAL frameworks by replicating their results. The empirical outcomes, as presented in Table 3, consistently establish the superior performance of our RobustSentEmbed framework in contrast to various other sentence embeddings. Our framework achieves the highest average Spearman's correlation score when compared to state-of-the-art approaches. Specifically, utilizing the BERT encoder, our framework surpasses the second-best embedding method, USCAL, by a margin of 1.59%. Moreover, RobustSentEmbed achieves the highest score in the majority of individual STS tasks, outperforming other embedding methods in 6 out of 7 tasks. Moreover, Our framework exhibits marginally enhanced performance in comparison to our prior RobustEmbed framework. For the RoBERTa encoder, RobustSentEmbed outperforms the state-of-the-art embeddings in five out of seven STS tasks and attains the highest average Spearman's correlation score.

4.4 Transfer Tasks

We leveraged transfer tasks to assess the performance of our framework, RobustSentEmbed, across a diverse range of text classification tasks, including sentiment analysis and paraphrase identification. Our evaluation encompassed six transfer tasks: CR (Hu and Liu, 2004), SUBJ (Pang and Lee, 2004), MPQA (Wiebe et al., 2005), SST2 (Socher et al., 2013), and MRPC (Dolan and Brockett, 2005). We trained a logistic regression classifier on top of the fixed sentence embeddings. To ensure the reliability of our findings, we replicated the SimCSE, ConSERT, and USCAL frameworks. The outcomes, as presented in Table 4, demonstrate the superior performance of our framework in terms of average accuracy when compared to other sentence embeddings. Specifically, when utilizing the BERT encoder, our framework outperforms the second-best embedding method by a margin of 0.23%. Furthermore, RobustSentEmbed achieves the highest score in four out of six text classification tasks. Our framework also achieves similar performance compared to our prior RobustEmbed framework. A similar trend is observed for the RoBERTa encoder. Overall, based on the results presented in Tables 3 and 4, we conclude that RobustSentEmbed generates general sentence representation in addition to robust representation (4.1 and section 4.2).

In conclusion, the comprehensive experiments, as indicated by the outcomes in Tables 1, 2, 3, and 4, along with Figure 2, confirm the exceptional performance of RobustSentEmbed in text representation and resilience against adversarial attacks and adversarial tasks. These findings highlight the framework's outstanding robustness and generalization capabilities, underscoring its potential as a versatile method for generating high-quality sentence embeddings.

4.5 Distribution of Sentence Embeddings

We employed two critical metrics, *alignment* and *uniformity* (Wang and Isola, 2020), for evaluating the quality of our representations. With a distribution of positive pairs p_{pos}, *alignment* computes the expected distance between the embeddings of paired instances:

$$\ell_{\text{align}} \triangleq \mathop{\mathbb{E}}_{(x,x^+)\sim p_{pos}} \|f(x) - f(x^+)\|^2 \tag{11}$$

Model	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
GloVe embeddings (avg.) ♡	55.14	70.66	59.73	68.25	63.66	58.02	53.76	61.32
$\text{BERT}_{\text{base}}$ (first-last avg.) ♣	39.70	59.38	49.67	66.03	66.19	53.87	62.06	56.70
$\text{BERT}_{\text{base}}$-flow ♣	58.40	67.10	60.85	75.16	71.22	68.66	64.47	66.55
$\text{BERT}_{\text{base}}$-whitening ♣	57.83	66.90	60.90	75.08	71.31	68.24	63.73	66.28
ConSERT-$\text{BERT}_{\text{base}}$	64.56	78.55	69.16	79.74	76.00	73.91	67.35	72.75
ATCL-$\text{BERT}_{\text{base}}$	67.14	80.86	71.73	79.50	76.72	79.31	70.49	75.11
SimCSE-$\text{BERT}_{\text{base}}$	68.66	**81.73**	72.04	80.53	78.09	79.94	71.42	76.06
USCAL-$\text{BERT}_{\text{base}}$	69.30	80.85	72.19	81.04	77.52	81.28	71.98	76.31
RobustEmbed-$\text{BERT}_{\text{base}}$	70.52	82.13	73.56	82.38	77.72	82.97	73.24	77.51
RobustSentEmbed-$\text{BERT}_{\text{base}}$	**71.90**	81.12	**74.92**	**82.38**	**79.43**	**82.02**	**73.53**	**77.90**
$\text{RoBERTa}_{\text{base}}$-whitening	46.99	63.24	57.23	71.36	68.99	61.36	62.91	61.73
ConSERT-$\text{RoBERTa}_{\text{base}}$	66.90	79.31	70.33	80.57	77.95	81.42	68.16	74.95
SimCSE-$\text{RoBERTa}_{\text{base}}$	68.75	80.81	71.19	81.79	79.35	82.62	69.56	76.30
USCAL-$\text{RoBERTa}_{\text{base}}$	69.28	81.15	72.81	81.47	**80.55**	83.34	70.94	77.08
RobustEmbed-$\text{RoBERTa}_{\text{base}}$	69.71	81.77	73.34	81.98	79.74	83.70	71.10	77.33
RobustSentEmbed-$\text{RoBERTa}_{\text{base}}$	**70.03**	**82.15**	**73.27**	**82.48**	79.61	**83.82**	**71.66**	**77.57**
USCAL-$\text{RoBERTa}_{\text{large}}$	68.70	**81.84**	74.26	82.52	**80.01**	83.14	76.30	78.11
RobustEmbed-$\text{RoBERTa}_{\text{large}}$	68.92	81.53	74.35	82.91	79.98	83.93	76.93	78.36
RobustSentEmbed-$\text{RoBERTa}_{\text{large}}$	**69.30**	81.76	**75.14**	**83.57**	79.74	**83.90**	**77.08**	**78.64**

Table 3: Semantic Similarity performance on STS tasks (Spearman's correlation, "all" setting) for sentence embedding models. We emphasize the top-performing numbers among models that share the same pre-trained encoder. ♡: results from Reimers and Gurevych (2019); ♣: results from (Gao et al., 2021); All remaining results have been reproduced and reevaluated by our team. RobustSentEmbed produces the most effective sentence representations that are more general in addition to robust representation (section 4.2 and 4.1).

Model	MR	CR	SUBJ	MPQA	SST2	MRPC	Avg.
GloVe embeddings (avg.) ♣	77.25	78.30	91.17	87.85	80.18	72.87	81.27
Skip-thought ♡	76.50	80.10	93.60	87.10	82.00	73.00	82.05
BERT-`[CLS]` embedding ♣	78.68	84.85	94.21	88.23	84.13	71.13	83.54
ConSERT-$\text{BERT}_{\text{base}}$	79.52	87.05	94.32	88.47	85.46	72.54	84.56
SimCSE-$\text{BERT}_{\text{base}}$	81.29	86.94	94.72	89.49	**86.70**	75.13	85.71
USCAL-$\text{BERT}_{\text{base}}$	81.54	**87.12**	95.24	89.34	85.71	75.84	85.80
RobustEmbed-$\text{BERT}_{\text{base}}$	81.94	87.45	95.04	89.88	86.47	76.40	86.20
RobustSentEmbed-$\text{BERT}_{\text{base}}$	**82.06**	86.28	**95.42**	**89.61**	86.12	**76.69**	**86.03**
SimCSE-$\text{RoBERTa}_{\text{base}}$	81.15	87.15	92.38	86.79	**86.24**	75.49	84.87
USCAL-$\text{RoBERTa}_{\text{base}}$	**82.15**	87.22	92.76	87.74	84.39	76.20	85.08
RobustEmbed-$\text{RoBERTa}_{\text{base}}$	81.49	87.54	93.37	87.95	84.63	76.62	85.27
RobustSentEmbed-$\text{RoBERTa}_{\text{base}}$	81.57	**87.66**	**93.51**	**87.94**	85.04	**76.89**	**85.44**
USCAL-$\text{RoBERTa}_{\text{large}}$	**82.84**	87.97	93.12	88.48	**86.28**	76.41	85.85
RobustEmbed-$\text{RoBERTa}_{\text{large}}$	82.38	88.27	93.91	88.79	86.01	77.11	86.08
RobustSentEmbed-$\text{RoBERTa}_{\text{large}}$	82.56	**88.51**	**93.84**	**88.65**	86.18	**77.01**	**86.13**

Table 4: Results of transfer tasks for different sentence embedding models. ♣: results from Reimers and Gurevych (2019); ♡: results from Zhang et al. (2020); We emphasize the top-performing numbers among models that share the same pre-trained encoder. All remaining results have been reproduced and reevaluated by our team. RobustSentEmbed outperforms all other methods, regardless of the pre-trained language model ($\text{BERT}_{\text{base}}$, $\text{RoBERTa}_{\text{base}}$, or $\text{RoBERTa}_{\text{large}}$).

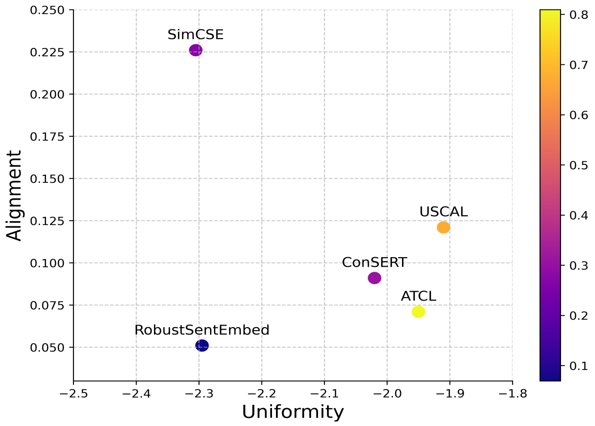

Figure 3: $\ell_{\text{align}} - \ell_{\text{uniform}}$ plot of models based on $\text{BERT}_{\text{base}}$. Lower uniformity and alignment is better.

Uniformity measures how well the embeddings are uniformly distributed in the representation space:

$$\ell_{\text{uniform}} \triangleq \log \mathop{\mathbb{E}}_{x,y \overset{i.i.d.}{\sim} p_{\text{data}}} e^{-2\|f(x)-f(y)\|^2} \quad (12)$$

Figure 3 shows the *uniformity* and *alignment* of different sentence embedding models. Smaller values indicate better performance. In comparison to the other representations, RobustSentEmbed achieves a similar level of *uniformity* (-2.295 vs. -2.305) but exhibits superior *alignment* (0.051 vs. 0.073). This demonstrates that our framework is more efficient in optimizing the representation space in two different directions.

5 Conclusion and Future Work

This paper introduces RobustSentEmbed, a self-supervised sentence embedding framework enhancing robustness against adversarial attacks while achieving state-of-the-art performance in text representation and NLP tasks. Current sentence embeddings are vulnerable to attacks, and RobustSentEmbed addresses this by generating high-risk perturbations at token and sentence levels. These perturbations are incorporated into novel contrastive and difference prediction objectives. The framework is validated through comprehensive experiments on semantic textual similarity and transfer learning tasks, confirming its robustness against adversarial attacks and semantic similarity tasks. In future research, we aim to investigate the use of hard negative examples to further enhance the effectiveness of text representations.

6 Limitations

Despite the effectiveness of our approach and its notable performance, there are potential limitations to our framework:

- The framework is primarily tailored for descriptive models like BERT, adept at language understanding and representation, including tasks such as text classification. However, its direct application to generative models like GPT, focused on generating coherent and contextually relevant text, may pose challenges. Thus, applying our methodology to enhance generalization and robustness in generative pre-trained models might have limitations.

- Utilizing substantial GPU resources is necessary for pre-training large-scale models like $\text{RoBERTa}_{\text{large}}$ in our framework. Due to limited GPU availability, we had to use smaller batch sizes during pre-training. Although larger batch sizes typically result in better performance, our experiments had to compromise and use smaller batch sizes to efficiently generate sentence embeddings within GPU constraints.

Acknowledgments

This work was supported in part by the National Science Foundation under Grants Nos. 2244219, 2315596, 2146497, 2413654, and 2413856.

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Inigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, et al. 2015. Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pages 252–263.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. SemEval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 81–91, Dublin, Ireland. Association for Computational Linguistics.

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. SemEval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 497–511, San Diego, California. Association for Computational Linguistics.

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 task 6: A pilot on semantic textual similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393, Montréal, Canada. Association for Computational Linguistics.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *SEM 2013 shared task: Semantic textual similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 32–43, Atlanta, Georgia, USA. Association for Computational Linguistics.

Javad Asl, Eduardo Blanco, and Daniel Takabi. 2023. RobustEmbed: Robust sentence embeddings using self-supervised contrastive pre-training. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4587–4603, Singapore. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

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.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*.

Alexis Conneau and Douwe Kiela. 2018. SentEval: An evaluation toolkit for universal sentence representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. 2023. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5(3):220–235.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Siddhant Garg and Goutham Ramakrishnan. 2020. BAE: BERT-based adversarial examples for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6174–6181, Online. Association for Computational Linguistics.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Jens Hauser, Zhao Meng, Damian Pascual, and Roger Wattenhofer. 2023. Bert is robust! a case against word substitution-based adversarial attacks. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177.

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.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020a. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130, Online. Association for Computational Linguistics.

Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2019. TextBugger: Generating adversarial text against real-world applications. In *Proceedings 2019 Network and Distributed System Security Symposium*. Internet Society.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020b. BERT-ATTACK: Adversarial attack against BERT using BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6193–6202, Online. Association for Computational Linguistics.

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.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A sick cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 216–223.

Deshui Miao, Jiaqi Zhang, Wenbo Xie, Jian Song, Xin Li, Lijuan Jia, and Ning Guo. 2021. Simple contrastive representation adversarial learning for nlp tasks.

John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126.

Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas Tezak, Jong Wook Kim, Chris Hallacy, Johannes Heidecke, Pranav Shyam, Boris Power, Tyna Eloundou Nekoul, Girish Sastry, Gretchen Krueger, David Schnurr, Felipe Petroski Such, Kenny Hsu, Madeleine Thompson, Tabarak Khan, Toki Sherbakov, Joanne Jang, Peter Welinder, and Lilian Weng. 2022. Text and code embeddings by contrastive pre-training.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

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. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Annual Meeting of the Association for Computational Linguistics*.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 115–124, Ann Arbor, Michigan. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Conference on Empirical Methods in Natural Language Processing*.

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, Florence, Italy. Association for Computational Linguistics.

Daniela N. Rima, DongNyeong Heo, and Heeyoul Choi. 2022. Adversarial training with contrastive learning in nlp. *Computer Speech & Language*. Submitted.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. 2021. Whitening sentence representations for better semantics and faster retrieval. *CoRR*, abs/2103.15316.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *China national conference on Chinese computational linguistics*, pages 194–206. Springer.

Dong Wang, Ning Ding, Piji Li, and Hai-Tao Zheng. 2021. Cline: Contrastive learning with semantic negative examples for natural language understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 2332–2342. Association for Computational Linguistics.

Qian Wang, Weiqi Zhang, Tianyi Lei, Yu Cao, Dezhong Peng, and Xu Wang. 2023. Clsep: Contrastive learning of sentence embedding with prompt. *Knowledge-Based Systems*, 266:110381.

Tongzhou Wang and Phillip Isola. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pages 9929–9939. PMLR.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language resources and evaluation*, 39(2):165–210.

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

Chen Wu, Ruqing Zhang, Jiafeng Guo, Maarten De Rijke, Yixing Fan, and Xueqi Cheng. 2023. Prada: Practical black-box adversarial attacks against neural ranking models. *ACM Transactions on Information Systems*, 41(4):1–27.

Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021. ConSERT: A contrastive framework for self-supervised sentence representation transfer. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5065–5075, Online. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28:649–657.

Yan Zhang, Ruidan He, Zuozhu Liu, Kwan Hui Lim, and Lidong Bing. 2020. An unsupervised sentence embedding method by mutual information maximization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 1601–1610. Association for Computational Linguistics.

A Training Details

we initialize our sentence encoder using the checkpoints obtained from BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019). RobustSentEmbed utilizes the representation of the [CLS] token as the starting point and incorporates a pooler layer on top of the [CLS] representations to facilitate contrastive learning objectives. The training process of RobustSentEmbed involves 4 epochs. The best checkpoint, determined by the highest average STS score, is selected for final evaluation. To train the model, we utilize a dataset consisting of 10^6 randomly sampled sentences from English Wikipedia, as provided by the SimCSE framework (Gao et al., 2021). The average training time for RobustSentEmbed is 2-4 hours. As our framework is initialized with pre-trained checkpoints, it exhibits robustness that is not sensitive to batch sizes, thus enabling us to employ batch sizes of either 64 or 128.

B Ablation Studies

In this section, we conduct an analysis of the impact of five critical hyperparameters employed in the RobustSentEmbed framework on its overall performance. $\text{BERT}_{\text{base}}$ is employed as the encoder,

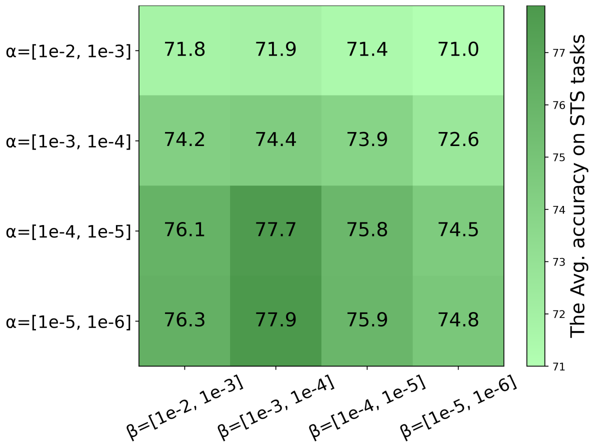

Figure 4: The impact of step sizes in perturbation generation on the average performance of STS tasks.

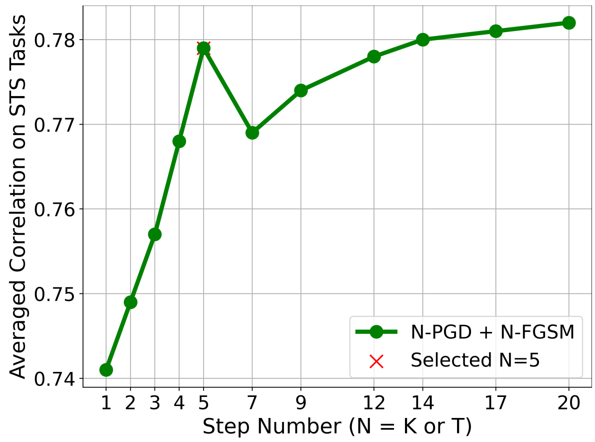

Figure 5: The impact of the step number (represented by N = K or T) in the T-step FGSM and K-step PGD methods on the averaged correlation of the STS tasks.

and the assessment of hyperparameters is carried out using the development set of STS tasks.

B.1 Step Sizes in Perturbation Generator

The RobustSentEmbed framework integrates two step sizes, denoted as α and β, to conduct iterative updates during the PGD and FGSM perturbation generation processes, respectively. Figure 4 shows the cooperative impact of adjusting the ranges for these two step sizes in generating high-risk perturbations, a crucial aspect for achieving an effective contrastive learning objective. The outcomes demonstrate more substantial improvements when β is fine-tuned to a lower bound, coupled with α set to an upper bound. More precisely, enhanced performance is evident when α and β are allocated ranges of [1e-4, 1e-6] and [1e-3, 1e-4], respectively. Consequently, we employ α = 1e-5 and β = 1e-3 for our experiments, as this configuration yields the optimal results among the different configurations.

B.2 Step Numbers in Perturbation Generator

RobustSentEmbed employs T-step FGSM and K-step PGD iterations to acquire high-risk adversarial perturbations for the contrastive learning objective. For simplicity in perturbation generation analysis, we establish K = T. The influence of varying step numbers (N = K or T) on effectiveness is illustrated in Figure 5. A gradual improvement is observed as N increases from 1 to 12; however, beyond N=12, the improvement becomes negligible. Additionally, higher N results in longer running time and inequitable resource allocation. Consequently, we opt for N=5 in our experiments.

B.3 Norm Constraint

To ensure imperceptibility in the generated adversarial examples, RobustSentEmbed regulates the magnitude of the perturbation vectors (whether $\boldsymbol{\delta}$ or $\boldsymbol{\eta}$). This control is achieved through the utilization of three commonly employed norm functions: L_1, L_2, and L_∞, to restrict the magnitude of the perturbation to small values. The averaged Spearman's correlation of these norm functions across different Semantic Textual Similarity tasks is presented in Table 5. The L_∞ norm exhibits superior correlation in comparison to the other two norms, thus warranting its selection as the norm function for our experimental assessment.

Norm	Correlation
L_∞	**77.90**
$L2$	76.84
$L1$	76.52

Table 5: The impact of the norm constraint on perturbation generation on the average performance of various STS tasks.

B.4 Contrastive Learning Loss

The first part of the total loss function (Equation 10) is dedicated to optimizing the similarity between the input instance x and its positive pair (x^{pos}), as well as the similarity between x and its adversarial perturbation (x^{adv}). While this indirectly brings x^{pos} and x^{adv} closer, our findings indicate that incorporating direct contrastive learning between x^{pos} and x^{adv} (the second part of Equation 10) through the regularization of the objective

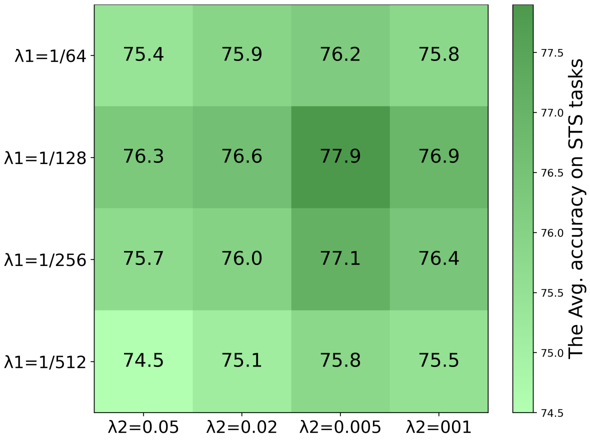

Figure 6: The impact of weighting coefficients in the total loss function on the average performance of STS tasks.

function in the first part helps us achieve enhanced clean accuracy and robustness. Additionally, the third part of the total loss function introduces the adversarial replaced token detection objective into the loss function, making it more challenging for adversarial training to converge. Figure 6 illustrates the impact of different values of the weighting coefficients (i.e., λ_1, λ_2) on the final performance of our framework. As illustrated, when $\lambda_1 = 1/128$ and $\lambda_2 = 0.005$, the framework achieves the highest average accuracy for semantic textual similarity tasks. We utilize $\lambda_1 = 1/128$ and $\lambda_2 = 0.005$ for all other experiments.

B.5 Modulation Factor

RobustSentEmbed includes a modulation factor, represented as $0 \leq \rho \leq 1$, to adjust the relative importance of each individual perturbation (PGD and FGSM) in the formation of the sentence-level perturbation. The efficacy of different values of this modulation factor on semantic textual similarity tasks is detailed in Table 6. The findings reveal that $\rho = 0.5$ yields the highest averaged correlation across the examined magnitudes, underscoring its capability to generate more powerful perturbations. Consequently, we employ this configuration in the setup of our framework.

C Adversarial Attack Methods

This section provides additional details regarding the various adversarial attacks. The TextBugger method (Li et al., 2019) identifies crucial words by analyzing the Jacobian matrix of the target model and selects the optimal perturbation from a set of five generated perturbations. The PWWS (Ren et al., 2019) employs a synonym-swap technique based on a combination of word saliency scores and maximum word-swap effectiveness. TextFooler (Jin et al., 2020) identifies significant words, gathers synonyms, and replaces each such word with the most semantically similar and grammatically correct synonym. The BAE (Garg and Ramakrishnan, 2020) employs four adversarial attack strategies involving word replacement and/or word insertion operations to generate substitutions. The BERTAttack (Li et al., 2020b) comprises two steps: (a) identifying vulnerable words/sub-words and (b) utilizing BERT MLM to generate semantic-preserving substitutes for the vulnerable tokens.

ρ	**Correlation**
0	76.06
0.25	76.85
0.5	**77.90**
0.75	77.34
1	76.34

Table 6: The impact of the modulation factor on the average performance of different Semantic Textual Similarity (STS) tasks in generating the final perturbation.

D RobustSentEmbed Algorithm

Algorithm 1 illustrates our framework's approach to generating a norm-bounded perturbation at both the token-level and sentence-level using an iterative process. It confuses the $f_\theta(\cdot)$ encoder by treating the perturbed embeddings as different instances. Our framework then utilizes a contrastive learning objective in conjunction with a replaced token detection objective to maximize the similarity between the embedding of the input sentence and the adversarial embedding of its positive pair (former objective), as well as its edited sentence (latter objective).

Algorithm 1: RobustSentEmbed Algorithm

Input: Epoch number E, PLM Encoder $\mathbf{f}_{\boldsymbol{\theta}}$, dataset of raw sentences $\boldsymbol{\mathcal{D}}$, embedding perturbation $\{\boldsymbol{\delta}, \boldsymbol{\eta}\}$, dropout masks m_1 and m_2, perturbation bound ϵ, adversarial step sizes $\{\alpha, \beta, \gamma\}$, learning rate ξ, perturbation modulator ρ, weighting coefficients $\{\lambda_1, \lambda_2\}$, adversarial steps $\{K, T\}$, contrastive learning objective $\mathcal{L}_{con,\theta}$ (eq. 9), ELECTRA generator $G(.)$ and discriminator $D(.)$

Output: Robust Sentence Representation

$\mathcal{V} \in \mathbb{R}^{N*D} \leftarrow \frac{1}{\sqrt{D}}\mathrm{U}(-\sigma, \sigma)$

for epoch $= 1, ..., E$ **do**

  for *minibatch* $\mathrm{B} \subset \boldsymbol{\mathcal{D}}$ **do**

    $\boldsymbol{\delta}^0 \leftarrow \frac{1}{\sqrt{D}}\mathrm{U}(-\sigma, \sigma)$, $\boldsymbol{\eta}_i^0 \leftarrow \mathcal{V}[w_i]$

    $\boldsymbol{X} \ = \mathbf{f}_{\boldsymbol{\theta}}$.embedding(B, m_1)

    $\boldsymbol{X}^+ = \mathbf{f}_{\boldsymbol{\theta}}$.embedding(B, m_2)

    for t = $1, ..., max(K,\ T)$ **do**

      $\boldsymbol{g}_\delta = \nabla_\delta \mathcal{L}_{con,\theta}(\boldsymbol{X} + \boldsymbol{\delta}^{t-1} + \boldsymbol{\eta}^{t-1}, \{\boldsymbol{X}^+\})$

      if $t \leq K$ **then**

        $\boldsymbol{\delta}^t_{pgd} = \Pi_{\|\boldsymbol{\delta}\|_P \leq \epsilon}(\boldsymbol{\delta}^{t-1} + \alpha g(\boldsymbol{\delta}^{t-1}) / \|g(\boldsymbol{\delta}^{t-1})\|_P)$

      end

      if $t \leq T$ **then**

        $\boldsymbol{\delta}^t_{fgsm} = \Pi_{\|\boldsymbol{\delta}\|_P \leq \epsilon}(\boldsymbol{\delta}^{t-1} + \beta \mathrm{sign}(g(\boldsymbol{\delta}^{t-1})))$

      end

      $\boldsymbol{g}_{\eta_i} = \nabla_\eta \mathcal{L}_{con,\theta}(\boldsymbol{X} + \boldsymbol{\delta}^{t-1} + \boldsymbol{\eta}^{t-1}, \{\boldsymbol{X}^+\})$

      $\boldsymbol{\eta}^t_i = n^i * (\boldsymbol{\eta}^t_{i-1} + \gamma \boldsymbol{g}_{\eta_i} / \|\boldsymbol{g}_{\eta_i}\|_P)$

      $\boldsymbol{\eta}^t \leftarrow \Pi_{\|\boldsymbol{\eta}\|_P \leq \epsilon}(\boldsymbol{\eta}^t)$

    end

    $\mathcal{V}[w_i] \leftarrow \boldsymbol{\eta}_i^{max(K,\, T)}$

    $\boldsymbol{\delta}_f = \rho \boldsymbol{\delta}^K_{pgd} + (1 - \rho)\boldsymbol{\delta}^T_{fgsm}$

    for $x \in \mathrm{B}$ **do**

      $x^{''} = G(\mathrm{MLM}(x))$

      $X^{adv} = X^{''} + \boldsymbol{\eta}_i^{max(K,\, T)}$

      $\mathcal{L}^x_{RTD,\,\theta} = \sum_{j=1}^{|x|}[-\mathbb{1}(X^{adv}_j = X_j) \log D(X^{adv},\, \mathbf{f}_{\boldsymbol{\theta}}(x),\, j)$
      $-\mathbb{1}(X^{adv}_j \neq X_j) \log (1 - D(X^{adv},\, \mathbf{f}_{\boldsymbol{\theta}}(x),\, j))]$

    end

    $\mathcal{L}_{RTD,\,\theta} = \sum_{i=1}^{|B|} \mathcal{L}^{x_i}_{RTD}$

    $\mathcal{L}_{RobustEmbed,\,\theta} := \mathcal{L}_{con,\,\theta}(\boldsymbol{X}, \{\boldsymbol{X}^+,\ \boldsymbol{X} + \boldsymbol{\delta}_f\})$

    $$\mathcal{L}_{\text{total}} := \mathcal{L}_{\text{RobustEmbed},\theta} + \lambda_1 \cdot \mathcal{L}_{\text{con},\theta}(\boldsymbol{X} + \boldsymbol{\delta}_f, \{\boldsymbol{X}^+\}) + \lambda_2 \cdot \mathcal{L}_{\text{RTD},\theta}$$

    $\theta = \theta - \xi \nabla_{\boldsymbol{\theta}} \mathcal{L}_{total}$

  end

end