V-advCSE: Virtual Adversarial Contrastive Learning for Sentence Embeddings

Kyung Min Ko
School of Electrical and Computer Engineering, Purdue University
ko120@purdue.edu

Abstract

Learning sentence embedding is a fundamental problem in the natural language processing domain. Many state-of-the-art models are pre-trained on large corpora and then fine-tuned to solve downstream tasks. However, current large language models are vulnerable to adversarial attacks due to aggressive fine-tuning on pre-trained models, which compromises their ability to generalize to unseen data. In this paper, we propose V-advCSE (Virtual adversarial contrastive learning for sentence embeddings), an unsupervised contrastive learning framework that adopts a virtual adversarial training approach for sentence embeddings. Instead of using adversarial training, which requires labels for generating adversarial examples, we leveraged a virtual adversarial training framework that does not require labels to generate adversarial examples. It applies perturbations in the embedding space using projected gradient ascent methods. This maximizes the difference within the model's output distribution in the constraint region, ensuring no significant change in the overall output distribution. We evaluated our V-advCSE on both STS (semantic textual similarity) and ANLI (adversarial natural language inference) tasks to test performance on both general and adversarial datasets. Our results show that V-advCSE outperformed SimCSE by 1.64 points on semantic textual similarity (STS) tasks and 2.2 points on adversarial natural language inference (ANLI) tasks.

1 Introduction

Learning universal sentence representations that capture the essential semantics of a sentence, and leveraging them in downstream tasks, has been extensively studied in research (Gao et al., 2021; Kiros et al., 2015; Conneau et al., 2017; Chuang et al., 2022; Giorgi et al., 2020). Recent work has also shown that contrastive learning in the natural language processing domain can be effective for various fine-tuning tasks (Gao et al., 2021; Chuang et al., 2022; Le-Khac et al., 2020a; Wu et al., 2020). Contrastive learning applies various data augmentation techniques to generate positive pairs, training the model to have more similar data representations than negative pairs. Recent work, such as SimCSE, which uses simple dropout as a data augmentation method to generate positive pairs, has demonstrated better performance than more complex data augmentation methods, like modifying words directly (Gao et al., 2021).

Even though SimCSE achieves high performance by leveraging simple dropout augmentation, it still faces robustness issues when subjected to adversarial attacks. The study revealed that large language models are vulnerable to adversarial attacks, which are formed by applying small perturbations to the data (Goodfellow et al., 2014). During adversarial training, we leverage constrained norm gradients to add small perturbations to the original sample to obtain an adversarial example. However, employing gradient-based adversarial attacks in the natural language processing domain is not trivial, since the word token space is discrete. Previous research focused on methods such as substitution-based methods, which replace the original word with similar words to generate adversarial examples (Ebrahimi et al., 2017; Alzantot et al., 2018; Li et al., 2020b; Bao et al., 2021). However, these methods are computationally inefficient as finding substitution words requires high computation. Instead, we leveraged a virtual adversarial training framework to generate adversarial examples by applying gradient-based perturbations in the embedding space, which can improve the model's generalization (Miyato et al., 2018). Virtual adversarial training not only mitigates issues associated with the need for labels in adversarial training but also improves computational efficiency compared to the word substitution method.

We conducted a comprehensive evaluation of the V-advCSE model on seven standard semantic

textual similarity (STS) tasks (Agirre et al., 2016; Cer et al., 2017; Marelli et al., 2014) with transfer tasks(Pang and Lee, 2005; Hu and Liu, 2004; Pang and Lee, 2004; Wiebe et al., 2005; Socher et al., 2013; Voorhees and Tice, 2000; Dolan and Brockett, 2005). In the STS tasks, our V-adv CSE achieved 77.73% on average score of STS using the Bert-Base model, which is 1.64% higher than the SimCSE model. Furthermore, we conducted an experiment on the ANLI dataset (Nie et al., 2019) to evaluate the robustness of the model against adversarial attack. Our results show that V-advCSE is effective in improving the stability of the model against adversarial attacks, achieving [ ]%, which is []% higher than SimCSE.

2 Background and related work

2.1 Contrastive Learning

Contrastive learning has been studied extensively in the computer vision domain, leading to state-of-the-art results (Le-Khac et al., 2020b; He et al., 2020; Chuang et al., 2020; Tian et al., 2020). The core idea of contrastive learning is to learn representation by pulling the distances closer between the representations of the original image and its augmented version (positive pairs), while increasing the distance from other images (negative pairs). Studies in the computer vision domain have revealed that random cropping, flipping, and color jittering are effective data augmentation techniques for generating positive pairs (Oord et al., 2018). In NLP, contrastive learning has been explored in various contexts, such as pre-train zero-shot predictions (Rethmeier and Augenstein, 2023), text summarization (Duan et al., 2019), and language modeling (Logeswaran and Lee, 2018). However, applying contrastive learning in the natural language processing domain presents challenges due to the discrete nature of word representation. The most challenging aspect is selecting appropriate data augmentation techniques to generate positive pairs. Recent studies have revealed that aggressive data augmentation techniques, such as word cropping, deletion, and substitution for generating positive pairs, can harm the performance (Gao et al., 2021). Instead, SimCSE (Gao et al., 2021) used a simple dropout method to generate positive pairs, achieving state-of-the-art performance.

2.2 Adversarial Learning

We want to begin this section by clarifying the set of notations that we will going to use on explaining both on adversarial and virtual adversarial learning. Let $x \in R^I$ and $y \in Q$ respectively denote an input vector and an output label with I ,the input dimension while Q is label space. We denote $D_l = \left\{x_l^{(n)}, y_l^{(n)} \mid n = 1, \ldots, N_l\right\}$ to represent labeled dataset and $D_{ul} = \left\{x_{ul}^{(m)} \mid m = 1, \ldots, N_{ul}\right\}$ for unlabeled dataset. We denote output distribution $p(y|x, \theta)$ parameterized by θ. In addition, we use $\hat{\theta}$ to denote specific iteration step of parameters during training process.

Adversarial training is a novel regularization method designed to improve robustness against the worst-case perturbations to the original input (Goodfellow et al., 2014). The loss function of adversarial training is defined as in equations 1 and 2, which follows framework of Ensemble Agreement (Bachman et al., 2014). The goal of Ensemble Agreement is to train the model robust so that introducing random perturbation does not affect output too much. The F is a non-negative distance measuring function between two distributions, such as Kullback–Leibler divergence, r_{adv} is perturbation that maximizes the difference between original and perturbed state within p norm, q is true distribution, p is parametric model distribution, and ϵ is constant value that constraint the p norm value.

$$L_{adv}(x_l, \theta) = F[q(y|x_l), p(y|x_l + r_{adv}, \theta)] \quad (1)$$

$$r_{adv} = \arg \max_{r;\|r\|_p \leq \varepsilon} F[q(y|x_l), p(y|x_l + r, \theta)], \quad (2)$$

As indicated in the equations, the true label y is required to produce a prediction from the model, which frames adversarial training as a supervised learning task. The main objective of this loss function is to approximate the true distribution q with the model prediction p, ensuring robustness against the adversarial attack r_{adv}. Generally, it is challenging to obtain an exact adversarial perturbation r due to the complexity of neural networks (Goodfellow et al., 2014). Instead, Goodfellow et al., 2014 approximated r_{adv} using linear approximations, such as L_2 and L_∞ norms as shown on equation 3 and 4 respectively. While using linear approximation, we can use one hot vector $v(y; y_l)$ to approximate $q(y|x_l)$. We can compute $\nabla_x F[h(y; x), p(y|x, \theta)]$ by back propagation in neural neural networks. We

provided derivation of r_adv in the appendix A.

$$r_{adv} \approx \frac{g}{\|g\|_2}, g = \nabla_x F[v(y; y_l), p(y|x_l, \theta)] \quad (3)$$

$$r_{adv} \approx esig(g), \quad (4)$$

2.3 Virtual Adversarial Learning

Even though adversarial training is effective in improving the robustness of models in a supervised setting, it is not always feasible to obtain labels due to limited resources. Therefore, Miyato et al., 2018 proposed a virtual adversarial training framework that eliminates the need for labels, making it suitable for semi-supervised learning (Miyato et al., 2018). The key difference between adversarial training and virtual adversarial training is that virtual adversarial training replaces the true distribution q with the model prediction p. This replacement is feasible, especially if the number of samples is large, as p should be close to q according to the law of large numbers (Hsu and Robbins, 1947). The term "virtual" in virtual adversarial training comes from the concept of using virtual labels, which are generated by model predictions, instead of actual labels. By using a virtual label, the loss function becomes as described in equations 5 and 6. We are using x_* to represent either x_l labeled data or x_{ul} unlabeled data since full label information is not always available. F is a non-negative distance measuring function between distributions, r_{adv} is the perturbation, θ is the model parameter, and $\hat{\theta}$ is current estimate of the model parameter. The purpose of using $\hat{\theta}$ is to prevent the propagation of gradients during the generation of adversarial examples.

$$L_{vadv}(x_*, \theta) = F\left[p(y|x_*, \hat{\theta}), p(y|x_* + r_{vadv}, \theta)\right] \quad (5)$$

$$r_{vadv} = \arg\max_{r;\|r\|_p \leq \varepsilon} F\left[p(y|x_*, \hat{\theta}), p(y|x_* + r, \theta)\right], \quad (6)$$

3 Virtual Adversarial Contrastive Learning

Our approach combines the standard contrastive learning objective from SimCSE with the virtual adversarial training framework (Miyato et al., 2018; Gao et al., 2021). We follow the approach of unsupervised SimCSE (Gao et al., 2021) to generate the positive example $x+$ by applying a different dropout rate to each sentence in an unlabeled sentence $\{x_i\}_{i=1}^m$ as a form of data augmentation. We can simply generate the positive pair by passing into the encoder x_i twice to apply different dropout rate. Using the Bert encoder f, we can compute the embedding of the sentence, $h_i = f(x_i)$. Therefore, our contrastive loss function, illustrated in Equation 7, involves h and $h+$ as a positive pair. In this equation, τ represents the temperature hyper parameter, N is a mini-batch of N pairs, and sim stands for cosine similarity.

$$L_{cont}(h_i, \theta) = -\log \frac{e^{sim(h_i, h_i^+)/\tau}}{\sum_{j=1}^N e^{sim(h_i, h_j^+)/\tau}} \quad (7)$$

Moreover, our approach includes a virtual adversarial training objective to improve the model's robustness against adversarial attacks. Instead of using an adversarial training framework, which requires true label information, we leverage virtual labels produced by the model's predictions. Since we are focusing on unsupervised training, we do not need label information x_l. The virtual adversarial loss is shown as follows:

$$L_{vadv}(x_{ul}, \theta) = F\left[p(y|x_{ul}, \hat{\theta}), p(y|x_{ul} + r_{vadv}, \theta)\right] \quad (8)$$

$$r_{vadv} = \arg\max_{r;\|r\|_p \leq \varepsilon} F\left[p(y|x_{ul}, \hat{\theta}), p(y|x_{ul} + r, \theta)\right], \quad (9)$$

where F is a non negative distance measuring function between two distributions, p is the model prediction, r_{vadv} is perturbation, θ is model parameter, and $\hat{\theta}$ is a constant copy of model parameter. It is important to notice that during the adversarial example generation process, we are using a copy of model parameter $\hat{\theta}$ instead of model parameter θ to prevent gradient flow while constructing adversarial examples. To obtain the worst case perturbation that maximizes the distance between each distribution, we can leverage projected gradient ascent algorithm (Madry et al., 2017). Therefore our final objective function becomes as equation 10 where λ is weighting factor that controls the contribution of L_{vadv}

$$L = L_{cont} + \lambda * L_{vadv} \quad (10)$$

Algorithm 1 shows the detailed implementation of virtual adversarial training. Initially, we initialize the random perturbation with a normal distribution centered at 0 and with a variance of σ^2. From lines 5 to 8, we run K iterations of the projected gradient ascent algorithm to obtain the worst-case perturbation. Previous research has revealed that

Algorithm 1 Virtual Adversarial Loss Algorithm

1: **Input**: $X = \{(x_1), \ldots, (x_n)\}$: the sentence embeddings, $f(x;\theta)$: original model, $f(x;\hat{\theta})$: copy of the original model parameterized by $\hat{\theta}$, σ^2: the variance of the random initialization of perturbation r_{vadv}, ε: constraint bound, K: the number of iterations for perturbation estimation, η: the step size for updating perturbation, Π: the projection operation, F: Distribution distance measuring function.
2: **for all** $(x) \in X$ **do**
3: $\quad r_{vadv} \sim \mathcal{N}(0, \sigma^2)$
4: $\quad$ **for** $m = 1, \ldots, K$ **do**
5: $\quad\quad g_{adv} \leftarrow \nabla_{r_{vadv}} F(f(x;\hat{\theta}), f(x + r_{vadv};\theta))$
6: $\quad\quad r_{vadv} \leftarrow \Pi_{\|r_{vadv}\|_{normtype} \leq \varepsilon}(r_{vadv} + \eta g_{adv})$
7: $\quad$ **end for**
8: $\quad L_{vadv} \leftarrow F(f(x;\hat{\theta}), f(x + r_{vadv};\theta))$
9: **end for**
10: **Output**: L_{vadv}

one iteration is sufficient to achieve optimal performance; therefore, we set K to 1 for our experiment (Miyato et al., 2018). There are several options for the distribution distance measuring function F, such as KL divergence, symmetric KL divergence, and Jensen-Shannon divergence (Andriamanalimanana et al., 2019; Nielsen, 2020). Furthermore, various normalization types can be used to constrain the gradient on line 7, such as $L2$, and $L\infty$ norms. We conducted an extensive ablation study to analyze the contribution of each options. After finding the worst-case perturbation through K iterations, we add the perturbation to the original embedding and compute the virtual adversarial loss.

4 Experiment

4.1 Setup

In our experiment, we evaluated V-advCSE on seven semantic textual similarity (STS) tasks and various transfer tasks to analyze its impact on generalization. We followed the same settings as the unsupervised SimCSE (Gao et al., 2021), which did not use the STS training dataset during training to ensure a rigorous evaluation. Our V-advCSE was built on the SimCSE Pytorch implementation. We utilized BERT base (Devlin et al., 2018) and RoBERTa base (Liu et al., 2019) as encoders to generate sentence embedding. Additionally, we applied an MLP layer with batch normalization (Ioffe and Szegedy, 2015) on top of the [CLS] token representation to enhance model performance. The effects of batch normalization are detailed in the ablation study section. Furthermore, we assessed the model's robustness against adversarial attacks using the ANLI dataset (Nie et al., 2019).

4.2 Data

During unsupervised pre-training, we utilized 10^6 randomly sampled sentences from English Wikipedia, which aligns with the dataset used in unsupervised SimCSE (Gao et al., 2021). Our model's evaluation focused on generalization performance, assessed through STS tasks and transfer tasks, and its robustness against adversarial attacks, tested using ANLI tasks. The STS evaluation consist of seven semantic textual similarity tasks, including STS 2012-2016 (Agirre et al., 2016), the STS benchmark (Cer et al., 2017), and SICK-Relatedness (Marelli et al., 2014), without using their training datasets to maintain an unsupervised setting. The transfer tasks included seven text classification tasks from SentEval (Conneau and Kiela, 2018): MR (Pang and Lee, 2005), CR (Hu and Liu, 2004), SUBJ (Pang and Lee, 2004), MPQA (Wiebe et al., 2005), SST2 (Socher et al., 2013), TREC (Voorhees and Tice, 2000), and MRPC (Dolan and Brockett, 2005). For these tasks, we employed a logistic regression classifier trained on frozen sentence embeddings, following the default configuration from SentEval (Conneau and Kiela, 2018). Additionally, the robustness of our model against adversarial attacks was evaluated using the test and development sets of ANLI tasks (Nie et al., 2019).

4.3 Results

Baselines We compared our result with competitive unsupervised baselines including SimCSE (Gao et al., 2021), IS-BERT(Zhang et al., 2020), DeCLUTR (Giorgi et al., 2021), CT-BERT (Ye et al., 2023), SG-OPT (Kim et al., 2021), BERT-flow (Li et al., 2020a), BERT- whitening (Su et al., 2021),

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
IS-$\text{BERT}_{\text{base}}$	56.77	69.24	61.21	75.23	70.16	69.21	64.25	66.58
CT-$\text{BERT}_{\text{base}}$	61.63	76.80	68.47	77.50	76.48	74.31	69.19	72.05
SG-OPT-$\text{BERT}_{\text{base}}$	66.84	80.13	71.23	81.56	77.17	77.23	68.16	74.62
SimCSE-$\text{BERT}_{\text{base (reproduced)}}$	66.20	81.88	72.86	81.74	78.82	78.57	**72.58**	76.09
V-advCSE-$\text{BERT}_{\text{base}}$	**72.05**	**83.15**	**75.71**	**83.12**	**79.46**	**79.90**	70.75	**77.73**

Table 1: STS tasks results from different sentence embedding models .

Model	MR	CR	SUBJ	MPQA	SST‘	TREC	MRPC	Avg.
GloVe embeddings (avg.)	77.25	78.30	91.17	87.85	80.18	83.00	72.87	81.52
Skip-thought	76.50	80.10	93.60	87.10	82.00	92.20	73.00	83.50
Avg. BERT embeddings	78.66	86.25	94.37	88.66	84.40	92.80	69.54	84.94
BERT-[CLS]embedding	78.68	84.85	94.21	88.23	84.13	91.40	71.13	84.66
IS-$\text{BERT}_{\text{base}}$	81.09	87.18	94.96	88.75	85.96	**88.64**	74.24	85.83
SimCSE-$\text{BERT}_{\text{base}}$	80.97	86.12	94.81	89.06	**86.00**	88.00	73.57	85.50
V-advCSE-$\text{BERT}_{\text{base}}$	**82.11**	**86.56**	**94.96**	**89.08**	85.50	88.20	**75.71**	**86.01**

Table 2: Transfer tasks results from different sentence embedding models .

and averaged GloVe embeddings (Pennington et al., 2014).

Semantic Textual Similarity (STS) We evaluated various encoder models on 7 STS tasks. We followed the same setting from SimCSE (Gao et al., 2021) while evaluating these models on STS tasks. Therefore, we used no additional regressor, following "all" aggregation method, and using Spearman's correlation unit for evaluation (Gao et al., 2021). Our V-advCSE on $\text{Bert}_{\text{base}}$ model clearly outperforms other unsupervised encoder models as shown on table 1. Specifically, our model significantly outperforms SimCSE (Gao et al., 2021) with improving averaged Sperman's corelation from 76.09% to 77.73%.

Transfer Tasks We showed our result for 7 different transfer tasks on table 2. We follwed the same training procedure from SimCSE (Gao et al., 2021) with training logistic regression classifier on top of frozen sentence embeddings from the model. Our V-advCSE model improved agerage performance from SimCSE from 85.50% to 86.01%.

ANLI Tasks We further extended our experiment to analyze the effect of virtual adversarial training on improving the robustness against the adversarial attack. We used ANLI dataset that contains multiple rounds of adversarial tasks consisted of test and development set. We followed training setting from (Nie et al., 2019) with training on combination of MNLI, SNLI, ANLI, and FEVER dataset. As shown on the table 3, our V-advCSE improved average accuracy on development set from 46.2% to 48.4% and 46.8% to 48.6% on test set. Therefore, we can conclude that our V-advCSE improve the robustness against adversarial attack.

5 Ablation studies

In this section, we conduced extensive ablation studies to support our model's effectiveness. We used $\text{BERT}_{\text{base}}$ model with development set from STS-B and tranfer tasks.

Effect of Batch Normalization In SimCSE (Gao et al., 2021), the authors leverage a single linear layer followed by a tanh activation function to extract features for computing loss. In the computer vision domain, it is well-known to use a two-layer pooler with Batch Normalization (Ioffe and Szegedy, 2015; Chen et al., 2020). We have shown that using Batch Normalization in both SimCSE and V-advCSE improves performance on the STS-B and transfer tasks, as demonstrated in Table 4. We can clearly observe that applying Batch Normalization on V-advCSE result into best performance.

Choice of Different Distribution Distance Measuring Function There are several options for distance measuring function F from our al-

Method	Dev				Test			
	R1	R2	R3	All	R1	R2	R3	All
SimCSE-BERT$_{\text{BASE}}$	53.8	42.8	42.6	46.4	52.6	45.1	43.1	46.8
V-advCSE-BERT$_{\text{BASE}}$	55.1	44.4	45.6	**48.4**	54.4	47.2	44.1	**48.6**

Table 3: ANLI task results on both SimCSE and V-advCSE

gorithm. According to Miyato et al., 2018, they used KL divergence to measure the distance between each distribution. However, KL divergence is lack of symmetry compared to Jensen-Shannon (JS) divergence (Nielsen, 2020) and symmetric KL divergence (Andriamanalimanana et al., 2019):

$$F_{KL}(P \parallel Q) = \sum_x P(x) \log \left(\frac{P(x)}{Q(x)} \right) \quad (11)$$

$$F_{SKL}(P, Q) = \frac{1}{2}(F_{KL}(P \parallel Q) + F_{KL}(Q \parallel P)) \quad (12)$$

$$F_{JS}(P \parallel Q) = \frac{1}{2}F_{KL}(P \parallel M) + \frac{1}{2}F_{KL}(Q \parallel M) \quad (13)$$

$$M = \frac{1}{2}(P + Q) \quad (14)$$

	STS-B	Avg. transfer
V-advCSE		
w/ BatchNorm	**84.55**	**84.41**
w/o BatchNorm	83.52	83.84
SimCSE		
w/ BatchNorm	82.22	84.36
w/o BatchNorm	81.47	83.61

Table 4: Performance comparison of models with and without Batch Normalization.

Function	STS-B	Avg. transfer
KL	84.08	84.12
Sym KL	84.21	84.33
JS KL	**84.55**	**84.41**

Table 5: Different distance measuring function performance

Therefore, we took experiment to analyze the effect of different distance measuring functions. As shown on the table 5, we can observe that including symmetric property while measuring distance between two distribution improve the STS-B and Avg. transfer performance. Overall, JS KL obtained best performance.

λ	0	$1e^{-6}$	$5e^{-5}$	$1e^{-5}$
STS-B	82.22	**84.55**	84.10	83.52
λ	$5e^{-5}$	$1e^{-4}$	$5e^{-4}$	$1e^{-3}$
STS-B	84.36	83.24	84.11	83.46

Table 6: Development set result under different λ

Effect of λ In the section 3, we introduced λ to adjust the effect of virtual adversarial loss. Since contrastive object is relatively easier task compare to virtual adversarial task, scale of contrastive loss is 100 times smaller than the virtual adversarial loss. Therefore, we used smaller coefficient values to balance between each other. In the table 6, we used different set of λ values to find the best performing λ value. As a result, we found that $\lambda = 1e^{-6}$ is best performing value.

6 Conclusion

In this paper, we presented V-advCSE, a novel method to improve the robustness of the model by leveraging virtual adversarial training framework. Empirical results shows that V-advCSE not only improved the generalized performance on STS and transfer tasks, but also improve the robustness of the model on ANLI tasks. We also conducted extensive ablation studies considering effect of Batch Normalization, different distance measuring function, and λ values. Our work can be further extended on supervised task in the future.

References

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez Agirre, Rada Mihalcea, German Rigau Claramunt, and Janyce Wiebe. 2016. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *SemEval-2016. 10th International Workshop on Semantic Evaluation; 2016 Jun 16-17; San Diego, CA. Stroudsburg (PA): ACL; 2016. p. 497-511.* ACL (Association for Computational Linguistics).

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

Bruno Andriamanalimanana, Ali Tekeoglu, Korkut Bekiroglu, Saumendra Sengupta, Chen-Fu Chiang, Michael Reale, and Jorge Novillo. 2019. Symmetric kullback-leibler divergence of softmaxed distributions for anomaly scores. In *2019 IEEE Conference on Communications and Network Security (CNS)*, pages 1–6. IEEE.

Philip Bachman, Ouais Alsharif, and Doina Precup. 2014. Learning with pseudo-ensembles. *Advances in neural information processing systems*, 27.

Rongzhou Bao, Jiayi Wang, and Hai Zhao. 2021. Defending pre-trained language models from adversarial word substitutions without performance sacrifice.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation.

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.

Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka. 2020. Debiased contrastive learning. volume 33, pages 8765–8775.

Yung-Sung Chuang, Rumen Dangovski, Hongyin Luo, Yang Zhang, Shiyu Chang, Marin Soljačić, Shang-Wen Li, Wen-tau Yih, Yoon Kim, and James Glass. 2022. Diffcse: Difference-based contrastive learning for sentence embeddings.

Alexis Conneau and Douwe Kiela. 2018. Senteval: An evaluation toolkit for universal sentence representations.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data.

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

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

Xiangyu Duan, Hoongfei Yu, Mingming Yin, Min Zhang, Weihua Luo, and Yue Zhang. 2019. Contrastive attention mechanism for abstractive sentence summarization.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2017. Hotflip: White-box adversarial examples for text classification.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings.

John Giorgi, Osvald Nitski, Bo Wang, and Gary Bader. 2020. Declutr: Deep contrastive learning for unsupervised textual representations.

John Giorgi, Osvald Nitski, Bo Wang, and Gary Bader. 2021. DeCLUTR: Deep contrastive learning for unsupervised textual representations. 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 879–895, Online. Association for Computational Linguistics.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738.

Pao-Lu Hsu and Herbert Robbins. 1947. Complete convergence and the law of large numbers. volume 33, pages 25–31. National Acad Sciences.

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.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr.

Taeuk Kim, Kang Min Yoo, and Sang-goo Lee. 2021. Self-guided contrastive learning for BERT sentence representations. 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 2528–2540, Online. Association for Computational Linguistics.

Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. volume 28.

Phuc H Le-Khac, Graham Healy, and Alan F Smeaton. 2020a. Contrastive representation learning: A framework and review. volume 8, pages 193907–193934. IEEE.

Phuc H Le-Khac, Graham Healy, and Alan F Smeaton. 2020b. Contrastive representation learning: A framework and review. volume 8, pages 193907–193934. IEEE.

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.

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

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.

Lajanugen Logeswaran and Honglak Lee. 2018. An efficient framework for learning sentence representations.

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

Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli. 2014. Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, pages 1–8.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. 2018. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. volume 41, pages 1979–1993. IEEE.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2019. Adversarial nli: A new benchmark for natural language understanding.

Frank Nielsen. 2020. On a generalization of the jensen–shannon divergence and the jensen–shannon centroid. volume 22, page 221. MDPI.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. *arXiv preprint cs/0409058*.

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

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 Rethmeier and Isabelle Augenstein. 2023. A primer on contrastive pretraining in language processing: Methods, lessons learned, and perspectives. volume 55, pages 1–17. ACM New York, NY.

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. volume abs/2103.15316.

Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. 2020. What makes for good views for contrastive learning? volume 33, pages 6827–6839.

Ellen M. Voorhees and Dawn M. Tice. 2000. Building a question answering test collection. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, page 200–207, New York, NY, USA. Association for Computing Machinery.

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

Zhuofeng Wu, Sinong Wang, Jiatao Gu, Madian Khabsa, Fei Sun, and Hao Ma. 2020. Clear: Contrastive learning for sentence representation.

Chao Ye, Guoshan Lu, Haobo Wang, Liyao Li, Sai Wu, Gang Chen, and Junbo Zhao. 2023. CT-BERT: Learning better tabular representations through cross-table pre-training.

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 (EMNLP)*, pages 1601–1610, Online. Association for Computational Linguistics.

A Appendix

In this section, we provide derivation of r_{adv} in L_2 and L_∞ case using linearity assumption of the model. With this assumption, we can apply first order approximation of Taylor series to obtain linear approximation of loss function.

A.1 L_2 Norm derivation

Both attacks used in adversarial and virtual adversarial learning framework can be formalized as maximum loss attack, where we want to find the perturbation r that maximizes loss within radius ϵ. To derive the maximum loss attack using the ℓ_2 norm, we need to solve the following optimization problem:

$$\max_{\|r\|_2 \leq \epsilon} \ell(h_\theta(x+r), y)$$

where r is the perturbation we want to find, ϵ is the radius of the ℓ_2 norm ball, ℓ is the loss function, h_θ is the model, and (x, y) is the input-label pair.

For small perturbations, the loss function $\ell(h_\theta(x+r), y)$ can be approximated linearly around x using first order taylor serise:

$$\ell(h_\theta(x+r), y) \approx \ell(h_\theta(x), y) + \nabla_x \ell(h_\theta(x), y)^T r$$

Here, $\nabla_x \ell(h_\theta(x), y)$ is the gradient of the loss function with respect to the input x.

The optimization problem then becomes:

$$\max_{\|r\|_2 \leq \epsilon} \nabla_x \ell(h_\theta(x), y)^T r$$

Using the Cauchy-Schwarz inequality, we know that:

$$\nabla_x \ell(h_\theta(x), y)^T r \leq \|\nabla_x \ell(h_\theta(x), y)\|_2 \|r\|_2$$

Given the constraint $\|r\|_2 \leq \epsilon$, it give us upper bound:

$$\nabla_x \ell(h_\theta(x), y)^T r \leq \epsilon \|\nabla_x \ell(h_\theta(x), y)\|_2$$

We can find r that satisfies upper bound as follows: We first multiply each side by $\nabla_x \ell(h_\theta(x), y)$ to make it as constant value, so we can move it between inequality.

$$\nabla_x \ell(h_\theta(x), y)^T r \nabla_x \ell(h_\theta(x), y) \leq \epsilon \|\nabla_x \ell(h_\theta(x), y)\|_2 \nabla_x \ell(h_\theta(x), y) \tag{15}$$

Now, we can divide each side by $\|\nabla_x \ell(h_\theta(x), y)\|_2^2$.

$$r \leq \epsilon \frac{\nabla_x \ell(h_\theta(x), y)}{\|\nabla_x \ell(h_\theta(x), y)\|_2}$$

The maximum value of $\nabla_x \ell(h_\theta(x), y)^T r$ is achieved when r is aligned with the gradient $\nabla_x \ell(h_\theta(x), y)$:

$$r^* = \epsilon \frac{\nabla_x \ell(h_\theta(x), y)}{\|\nabla_x \ell(h_\theta(x), y)\|_2}$$

A.2 L_∞ Norm derivation

To derive the maximum loss attack using the ℓ_∞ norm, we need to solve the following optimization problem:

$$\max_{\|r\|_\infty \leq \epsilon} \ell(h_\theta(x+r), y)$$

By applying first order Taylor series, we can again obtain linear approximation of loss function, thus the optimization problem then becomes:

$$\max_{\|r\|_\infty \leq \epsilon} \nabla_x \ell(h_\theta(x), y)^T r$$

Using Hölder's inequality, we know that:

$$\nabla_x \ell(h_\theta(x), y)^T r \leq \|\nabla_x \ell(h_\theta(x), y)\|_1 \|r\|_\infty$$

Given the constraint $\|r\|_\infty \leq \epsilon$, it gives us an upper bound:

$$\nabla_x \ell(h_\theta(x), y)^T r \leq \epsilon \|\nabla_x \ell(h_\theta(x), y)\|_1$$

We can find r that satisfies the upper bound from the objective function. If we want to maximize the $\nabla_x \ell(h_\theta(x), y)^T r$, then there would be two different cases. In order to maximize the objective function, we must satisfy $\|r\|_\infty = |\epsilon|$ by given constraint. Therefore, if the $(\nabla_x \ell(h_\theta(x), y))$ is negative, we need to have $r = -\epsilon$, and $r = \epsilon$ for positive $(\nabla_x \ell(h_\theta(x), y))$. As a result, we can conclude that r achieves max loss as follows:

$$r^* = \epsilon \cdot sign(\nabla_x \ell(h_\theta(x), y))$$

We can verify by plugging back to the upper bound.

$$\nabla_x \ell(h_\theta(x), y)^T \epsilon \cdot sign(\nabla_x \ell(h_\theta(x), y)) \leq \epsilon \|\nabla_x \ell(h_\theta(x), y)\|_1$$

Expanding the left side:

$$\epsilon \nabla_x \ell(h_\theta(x), y)^T sign(\nabla_x \ell(h_\theta(x), y)) = \epsilon \sum_i \nabla_x \ell(h_\theta(x), y)_i \cdot sign(\nabla_x \ell(h_\theta(x), y)_i)$$

Since $sign(\nabla_x \ell(h_\theta(x), y)_i)$ is 1 for positive elements and -1 for negative elements:

$$\epsilon \sum_i |\nabla_x \ell(h_\theta(x), y)_i| = \epsilon \|\nabla_x \ell(h_\theta(x), y)\|_1$$

Thus, we can achive upper bound when using r^*:

$$\epsilon \|\nabla_x \ell(h_\theta(x), y)\|_1 \leq \epsilon \|\nabla_x \ell(h_\theta(x), y)\|_1$$