AdCSE: An Adversarial Method for Contrastive Learning of Sentence Embeddings

Renhao Li, Lei Duan(✉), Guicai Xie, Shan Xiao, and Weipeng Jiang

School of Computer Science, Sichuan University, Chengdu, China
{lirenhao,guicaixie,shanxiao,weipengjiang}@stu.scu.edu.cn,
leiduan@scu.edu.cn

Abstract. Due to the impressive results on semantic textual similarity (STS) tasks, unsupervised sentence embedding methods based on contrastive learning have attracted much attention from researchers. Most of these approaches focus on constructing high-quality positives, while only using other in-batch sentences for negatives which are insufficient for training accurate discriminative boundaries. In this paper, we demonstrate that high-quality negative representations introduced by adversarial training help to learn powerful sentence embeddings. We design a novel method named AdCSE for unsupervised sentence embedding. It consists of an untied dual-encoder backbone network for embedding positive sentence pairs and a group of negative adversaries for training hard negatives. These two parts of AdCSE compete against each other mutually in an adversarial way for contrastive learning, obtaining the most expressive sentence representations while achieving an equilibrium. Experiments on 7 STS tasks show the effectiveness of AdCSE. The superiority of AdCSE in constructing high-quality sentence embeddings is also validated by ablation studies and quality analysis of representations.

Keywords: Sentence embedding · Contrastive learning · Adversarial training

1 Introduction

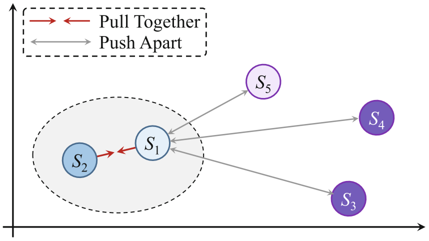
Sentence embeddings are used successfully for a variety of NLP applications such as semantic similarity comparison [10], sentence clustering [26], and information retrieval [23]. As a result, plenty of methods have been proposed and obtained high-quality sentence representations with additional supervision [8,16,25]. However, it is costly with human annotation and unavailable in real-world applications.

This work was supported in part by the National Key Research and Development Program of China (2018YFB0704301-1), the National Natural Science Foundation of China (61972268), the Sichuan Science and Technology Program (2020YFG0034).

A. Bhattacharya et al. (Eds.): DASFAA 2022, LNCS 13247, pp. 165–180, 2022.
https://doi.org/10.1007/978-3-031-00129-1_11

Sentences
S_1: A man is performing tricks on a motorcycle.
S_2: **A man is performing** acrobatics **on a motorcycle**.
S_3: Two dogs are running in the garden.
S_4: **A** woman **is** driving **a** car along the street.
S_5: The performer **is** tricking **a man on a motorcycle**.

(a) Sample sentences



(b) Embedding space

Fig. 1. A toy example of positives and negatives in contrastive learning where we bolded the same words as S_1 in the other sentences. S_1 is the input sentence; S_2 is a positive sentence of S_1 obtained by replacing synonyms; S_3, S_4 and S_5 are negatives of S_1, respectively. Among them, S_3 and S_4 are randomly sampled sentences while S_5 has a higher word coverage with S_1 compared to them. Thus, we take S_5 as a high-quality negative of S_1 for example.

Existing unsupervised sentence embedding methods with a contrastive objective have drawn much attention from researchers due to their impressive results on the semantic textual similarity (STS) tasks [6,12,17,29]. The main idea of contrastive learning is to pull semantically close neighbors (or 'positives') together and push apart non-neighbors (or 'negatives') [13].

How to construct positives and negatives for the given sentences is the key point of using contrastive learning in an unsupervised manner. Following this idea, recently developed methods, including ConSERT [29] and SimCSE [12], focus on constructing high-quality positives for the input sentences. ConSERT explores four different data augmentation strategies to generate positive views. SimCSE with an unsupervised manner applies the standard dropout twice as minimal data augmentation to compose positive pairs. For negatives, they only use all other sentences from the same batch where sentences are randomly sampled. This ignores that the quality of negatives also plays an important role in contrastive learning. Taking sentences in Fig. 1 as an example.

Example 1. It is easy for sentence embedding models to distinguish S_3 and S_4 as the negatives of S_1. However, S_5 is difficult to distinguish from S_1 due to its high word coverage with S_1. It is referred as a ***hard negative*** of S_1.

In general, hard negatives are more related to the input sentence in semantics compared to randomly sampled negatives. To build expressive sentence embeddings, we do not consider generating or sampling negative sentences from the input sentences. Instead, we directly obtain negative representations in embedding space by adversarial training.

In this paper, we design AdCSE: Adversarial Method for Contrastive Learning of Sentence Embeddings, which consists of an untied dual-encoder backbone network and a group of negative adversaries:

- *Backbone network*: most of the existing methods utilize two encoders with shared parameters as their backbone for embedding. Instead, we adopt an

untied dual-encoder as the backbone network to embed the input sentences and their corresponding positives.
- *Negative adversaries*: for unsupervised training hard negatives, negative adversaries are utilized to challenge the discriminative ability of backbone network by adversarial training.

With a contrastive learning objective, these two parts of AdCSE alternately update their parameters through adversarial training. When they reach equilibrium, the most expressive sentence embeddings will be obtained.

Our main contributions can be summarized as follows.

- We design a novel unsupervised method, named AdCSE, to build high-quality sentence embeddings with a contrastive learning objective.
- We improve the quality of negatives by introducing adversaries to an untied dual-encoder in contrastive learning framework. Expressive sentence embeddings are obtained by adversarial training between hard negatives and positives.
- We evaluate AdCSE on 7 STS tasks. Empirical results demonstrate the effectiveness of AdCSE over many competitive baselines. Additionally, fine-grained analysis such as embedding quality analysis and case study further validates its superiority in constructing powerful sentence embeddings.

The rest of this paper is organized as follows. Sect. 2 presents a comprehensive review of the related work. In Sect. 3, we discuss the critical techniques of the proposed model AdCSE. By comparing with many competitive baseline methods on STS tasks, the superior performance of AdCSE is demonstrated in Sect. 4. In Sect. 5, we get deep insight into AdCSE with further analysis. Finally, we conclude our work in Sect. 6.

2 Related Work

Our work is related to the existing research on sentence embedding and contrastive learning. We introduce the related work briefly below.

2.1 Sentence Embedding

Previous methods for sentence embedding include two main categories: (1) supervised learning with labeled sentences, and (2) unsupervised sentence embedding with unlabeled sentences, while a few of them adapt for both of the settings.

Supervised Approaches. To preserve the original information from sentences as much as possible, most of the early works focus on the fusion of multi-grained sentence features by CNNs or RNNs [8,16]. Since BERT [11] showed advanced performance on a variety of NLP downstream tasks, some attempts of generating sentence embedding using pre-trained language models have been applied to sentence-pair regression tasks. However, the native derived sentence representations from BERT are proved to be collapsed. To make full use of pre-trained language model in sentence-level tasks, Reimers *et al.* [25] first designed a sentence-BERT to derive semantically meaningful sentence embeddings. With siamese and triplet network structures, sentence-BERT is able to tackle semantic similarity search using cosine similarity.

Unsupervised Approaches. To further adapt sentence embeddings to downstream tasks like STS, a series of works are proposed for the anisotropy problem brought by BERT-based sentence representations. Li *et al.* [19] proposed a flow-based model by mapping embeddings to a standard Gaussian latent space. While BERT-whitening introduced by Su *et al.* [27] is another effective way to enhance the isotropy of sentence representations, which applies whitening operation to BERT and achieves competitive results. Another line of works are based on the distributional hypothesis (Mikolov *et al.* [22]), where context information of the sentences is considered adequately. For instance, Skip-thought (Kiros *et al.* [18]) utilizes an encoder-decoder framework to sequentially predict the words of adjacent sentences. Instead of training a model to reconstruct the surface form of the input sentence or its neighbors, Logeswaran *et al.* [20] designed quick thoughts (QT) to predict the adjacent sentences by the current sentence.

2.2 Contrastive Learning

Contrastive learning is a kind of self-supervised technique to learn powerful representation by distinguishing samples generated by the same object against the different. Based on this intuition, approaches with contrastive learning are enabled to achieve impressive results in unsupervised visual representation learning [9,14,15].

Recently, contrastive learning has been widely applied in NLP tasks for its strong ability to train the model in an unsupervised manner. Zhang *et al.* [30] proposed a CNN-based model IS-BERT, which constructs positive sample pairs by maximizing the mutual information between the global sentence embedding and its corresponding local contexts embeddings. Yan *et al.* [29] explored four kinds of data augmentation methods for sentence-level contrastive learning in both unsupervised and supervised settings. Instead of using a siamese network with shared parameters, Carlsson *et al.* [6] employed an untied dual-encoder framework to counter the task bias on final layers of models imposed by pre-training objectives. To make full use of embeddings of different layers in BERT, Kim *et al.* [17] designed a self-guided contrastive approach which fine-tunes the BERT by making the `[CLS]` representation of the last layer close to its hidden states. SimCSE proposed by Gao *et al.* [12] applies dropout to contrastive learning of sentence embeddings which acts as minimal data augmentation and performs effectively.

3 The Design of AdCSE

In this section, we present the details of AdCSE. We first introduce the problem formulation of unsupervised sentence embedding based on contrastive learning in Sect. 3.1. Then the backbone network for embedding positive sentence pairs

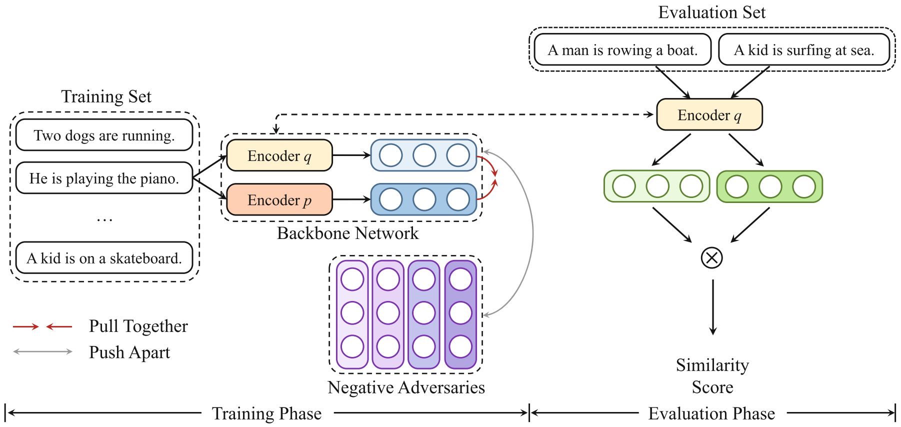


Fig. 2. Illustration of the proposed model AdCSE (Best viewed in color).

is presented in Sect. 3.2. In Sect. 3.3, we describe the negative adversaries for training hard negatives in detail. At last, the learning objective and algorithm of AdCSE are elaborated in Sect. 3.4. Figure 2 shows the architecture of AdCSE, which includes a dual-encoder backbone network along with the negative adversaries. These two parts of AdCSE interact with each other adversarially with a contrastive learning objective.

3.1 Problem Formulation

Given a set of input sentences $\mathcal{X}$, for each sentence $x_i \in \mathcal{X}$, the goal of unsupervised sentence embedding is to learn a representation $\mathbf{h}_i \in \mathbb{R}^d$ in embedding space $\mathcal{H}$. This process is abbreviated as $\mathcal{F} : \mathcal{X} \rightarrow \mathcal{H}$.

The goal of contrastive learning is to learn expressive representations by pulling semantically similar neighbors together and pushing non-neighbors apart. Specifically, for each sentence $x_i \in \mathcal{X}$, a function $\varphi(\cdot)$ is designed to map x_i to a semantically similar sentence $x_i^+ = \varphi(x_i)$ in order to compose a positive sentence pair (x_i, x_i^+). We adopt the normalized temperature-scaled cross-entropy loss (NT-Xent) as the contrastive objective. Denoting $\mathbf{h}_i \in \mathbb{R}^d$ and $\mathbf{h}_i^+ \in \mathbb{R}^d$ as the embeddings of x_i and x_i^+ mapped by the encoder in AdCSE, the training objective for (x_i, x_i^+) with a mini-batch of N sentence pairs is:

$$\mathcal{L}_i = -\log \frac{e^{\mathrm{sim}(\mathbf{h}_i, \mathbf{h}_i^+)/\tau}}{\sum_{j=1}^{N} e^{\mathrm{sim}(\mathbf{h}_i, \mathbf{h}_j^+)/\tau}} \tag{1}$$

where τ represents the temperature hyperparameter to control the scale of samples and $\mathrm{sim}(\cdot)$ calculates the cosine similarity of two embeddings.

3.2 Backbone Network for Contrastive Learning

To employ NT-Xent as the training objective in an unsupervised manner, some existing works construct the positive sample x_i^+ from the input sentence x_i with sentence-level data augmentation methods such as token shuffling, token cutoff [29] and dropout masks [12]. These methods utilize two encoders with shared parameters as their backbone to embed input sentences and their corresponding enhancements respectively, therefore may suffer from the semantic inconsistency issue between the input sentence and its positive. Instead, we follow He *et al.* [14] to utilize an untied dual-encoder as our backbone networks. In this case, we take advantage of two encoders with inconsistent parameters to distinguish between x_i and x_i^+ by themselves rather than an additional data augmentation process. In other words, for a given sentence x_i, we directly use the same sentence as its positive sample x_i^+.

In AdCSE, BERT is used for the two untied encoders: encoder q and encoder p. θ_q and θ_p are denoted as their corresponding learnable parameters. Then for a positive sentence pair (x_i, x_i^+), we apply two independent BERT encoders followed by pooling layers to map the sentences to representations $(\mathbf{h}_i, \mathbf{h}_i^+)$:

$$\mathbf{h}_i = g_{\theta_q}(f_{\theta_q}(x_i)) \tag{2}$$

$$\mathbf{h}_i^+ = g_{\theta_p}(f_{\theta_p}(x_i^+)) \tag{3}$$

where f_{θ_q} and f_{θ_p} are the `[CLS]` representations in the last layer of the two untied BERT. g_{θ_q} and g_{θ_p} stand for two independent pooling layers which consist of linear projection and the activation function $\tanh(\cdot)$. Note that θ_q and θ_p are updated in different ways in AdCSE for the dual-encoder to learn $\mathbf{h}_i$ and $\mathbf{h}_i^+$ respectively from the inconsistent parameters. To this end, gradient descent is adopted for optimizing θ_q while a momentum update is used to smooth the evolving process of θ_p which is proved to be effective by He *et al.* [14] in the field of computer vision. With $k \in \{1, 2, ..., K\}$ where K is the total steps of training, encoder parameters after the k-th step of training are denoted as $\theta_q^{(k)}$ and $\theta_p^{(k)}$. Given a momentum m, θ_p is updated as follows:

$$\theta_p^{(k)} = m\theta_p^{(k-1)} + (1-m)\theta_q^{(k-1)} \tag{4}$$

3.3 Adversaries for Hard Negatives Training

Inspired by Hu *et al.* [15], we adopt adversarial training to unsupervised construct hard negatives for contrastive sentence embedding. To challenge the ability of distinguishing between positives and negatives in the backbone network, negative adversaries $\mathcal{N} = \{\mathbf{n}_j | \mathbf{n}_j \in \mathbb{R}^d, 1 \leq j \leq M\}$ with M negatives are randomly initialized at first. Then they keep up with $\mathbf{h}_i$ in every sample batch by iteratively updating the learnable parameters of itself θ_n through adversarial training.

To be more specific, the backbone network tends to minimize the contrastive loss by making $\mathbf{h}_i$ close to $\mathbf{h}_i^+$ while pulling apart $\mathbf{h}_i$ and $\mathbf{n}_j$. In the meanwhile,

Algorithm 1. Pseudocode of AdCSE

```
Input: $\mathcal{X}$: training set; $K$: the total number of training steps; $\alpha_q$: learning rate of
    encoder $q$; $\alpha_n$: learning rate of adversaries; $m$: momentum for updating encoder $p$;
    $\tau$: temprature
Output: $\theta_q$: learning parameters of encoder $q$; $\theta_p$: learning parameters of encoder $p$;
    $\theta_n$: learning parameters of negative adversaries
initialize $\theta_q$, $\theta_p$ and $\theta_n$;
shuffle samples in $\mathcal{X}$;
for $k = 1 \rightarrow K$ do
    sample a batch $\mathcal{X}_{batch}^{(k)}$ from $\mathcal{X}$ without repetition;
    momentum update $\theta_p^{(k)}$ using Equation 4;
    for each sample $x_i$ in batch $\mathcal{X}_{batch}^{(k)}$ do
        obtain query embedding $\mathbf{h}_i$ by encoding $x_i$ with encoder $q$;
        obtain positive embedding $\mathbf{h}_i^+$ by encoding $x_i$ with encoder $p$;
    end for
    obtain embeddings of hard negatives in $\mathcal{N}$;
    compute the contrastive loss $\mathcal{L}$ using Equation 5;
    update $\theta_q^{(k)}$ using Equation 6;
    update $\theta_n^{(k)}$ using Equation 7;
end for
return $\theta_q$, $\theta_p$ and $\theta_n$
```

the negative adversaries tend to confuse the discriminator with the updated hard negatives by maximizing the contrastive loss. We believe the joint training of the backbone network and negative adversaries benefits the performance of AdCSE on evaluation.

3.4 Learning Objective and Algorithm

Based on the contrastive learning strategy, we present the following loss function of AdCSE which is derived from Equation 1:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{\mathrm{sim}(\mathbf{h}_i,\mathbf{h}_i^+)/\tau}}{e^{\mathrm{sim}(\mathbf{h}_i,\mathbf{h}_i^+)/\tau} + \sum_{j=1}^{M} e^{\mathrm{sim}(\mathbf{h}_i,\mathbf{n}_j)/\tau}} \tag{5}$$

where N is the number of positive sample pairs in a mini-batch while M is the number of adversaries for training hard negatives. Intuitively, the above objective tends to push the negative sample embedding $\mathbf{n}_j$ closer towards the input sentence embedding $\mathbf{h}_i$ from the current minibatch. Therefore, harder negatives will be trained by the adversaries with the parameters updated. In this way, gradient decent and ascent are respectively applied to update parameters θ_q and θ_n for adversarial training:

$$\theta_q^{(k)} = \theta_q^{(k-1)} - \alpha_q \frac{\partial \mathcal{L}(\theta_q^{(k-1)}, \theta_n^{(k-1)})}{\partial \theta_q^{(k-1)}} \tag{6}$$

$$\theta_n^{(k)} = \theta_n^{(k-1)} + \alpha_n \frac{\partial \mathcal{L}(\theta_q^{(k-1)}, \theta_n^{(k-1)})}{\partial \theta_n^{(k-1)}} \tag{7}$$

where α_q and α_n are learning rates of encoder q and the negative adversaries respectively. With the contrastive loss of the model $\mathcal{L}$ in Eq. 5, the adversarial target mentioned in Sect. 3.3 is presented as:

$$\theta_q^\star, \theta_n^\star = \arg\min_{\theta_q} \max_{\theta_n} \mathcal{L}(\theta_q, \theta_n) \tag{8}$$

where $\theta_q^\star$ and $\theta_n^\star$ are the parameters to equilibrate the two parts of AdCSE. We train this model in an adversarial way in the hope that the best performance of the model could be reached with the saddle point $(\theta_q^\star, \theta_n^\star)$ for this minimax problem. Based on the procedures above, we present the pseudo-code of AdCSE in Algorithm 1.

4 Experiments

We trained AdCSE on unlabeled Wikipedia corpus and evaluated its performance on 7 semantic textual similarity (STS) tasks. All experiments were conducted on a server with an RTX3090 and 24 GB memory. The model AdCSE was implemented by Python 3.6.2 with Pytorch 1.7.1 based on CUDA 11.0.

4.1 Experimental Setup

Datasets. Following Gao *et al.* [12], we used a million sentences randomly sampled from Wikipedia for our self-supervised training[1]. For evaluation, 7 STS tasks were utilized to conduct our experiments, including STS tasks 2012–2016 (STS12–STS16) (Agirre *et al.* [1–5], STS Benchmark (STS-B) (Cer *et al.* [7]) and SICK- Relatedness (SICK-R) (Marelli *et al.* [21]). Each sample in these datasets contains a pair of sentences together with a gold score between 0 and 5, indicating their ground-truth semantic similarities. We obtained all these datasets through the SentEval toolkit[2]. Please note that we only used development sets and test sets of STS tasks for evaluation so that all of the STS experiments were fully unsupervised.

Evaluation Metrics. We followed the evaluation metrics of SimCSE [12] to measure the semantic similarity of sentences. For sentence pairs in the evaluation set, we obtained their embeddings through $\mathcal{F}$ and calculated the set of

[1] https://huggingface.co/datasets/princeton-nlp/datasets-for-simcse/resolve/main/wiki1mforsimcse.txt.

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

predicted similarities $\hat{\mathcal{Y}}$ by cosine$(\cdot)$ function. Denoting the given set of ground-truth semantic scores as $\mathcal{Y}$, the ranks of $\hat{\mathcal{Y}}$ and $\mathcal{Y}$ are respectively acquired with the ranking function $\mathrm{r}(\cdot)$. To assess how well the relationship between these two variables are described using a monotonic function, the spearman correlation was applied to evaluate the correlation between them, which is defined as follows:

$$\rho = \frac{\mathrm{cov}(\mathrm{r}(\hat{\mathcal{Y}}), \mathrm{r}(\mathcal{Y}))}{\sigma_{\mathrm{r}(\hat{\mathcal{Y}})}\sigma_{\mathrm{r}(\mathcal{Y})}} \tag{9}$$

where cov$(\cdot)$ calculates the covariance of two variables while σ represents the standard deviations of them. The closer spearman correlation is to 1, the more similar the predicted ranked similarities from AdCSE and the ranked ground-truth are. To facilitate comparison with other baselines, we report $\rho \times 100$ as the spearman correlation in the rest of this paper.

Baselines. In our experiments, several state-of-the-art unsupervised sentence embedding methods were selected as baselines.

- **GloVe embeddings** [24] is an unsupervised learning algorithm to obtain vector representations for words. By performing aggregated global word-word co-occurrence statistics on a corpus, the method is able to generate sentence embeddings using the averaging word vectors.
- **BERT** [11] is a pre-trained language model using self-attention mechanism. Benifiting from both mask language model and next sentence prediction tasks, the model applies high-quality embeddings for various NLP tasks in a self-supervised manner.
- **BERT-flow** [19] maps embeddings to a standard Gaussian latent space to solve the anisotropy problem for sentence representations.
- **BERT-whitening** [27] enhances the isotropy of sentence representations by applying whitening operation to BERT.
- **IS-BERT** [30] is a CNN-based model which maximizes the mutual information to optimize sentence embeddings.
- **CT-BERT** [6] utilizes a dual-encoder framework together with contrastive loss to counter the task bias on final layers of models imposed by pre-training objectives.
- **ConSERT** [29] explores four kinds of data augmentation methods for sentence-level contrastive learning.
- **SG-BERT** [17] is a self-guided contrastive approach which fine-tunes the BERT by making the `[CLS]` output of the last layer close to its hidden states.
- **SimCSE** [12] applies dropout inside BERT as the minimal sentence-level data augmentation method in contrastive learning and acquires state-of-the-art performance on STS tasks.

Table 1. Evaluation results on the test set of STS tasks. We report the spearman correlation $\rho \times 100$ and bolded the best results. ♣: results from Reimers *et al.* [25]; ♢: results from Gao *et al.* [12]; ♡: results from Zhang *et al.* [30]; ♠: results from Yan *et al.* [29]; ★: results from Kim *et al.* [17]; baseline results without labels were implemented by ourselves.

Model	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
GloVe embeddings (avg.)$^{\clubsuit}$	55.14	70.66	59.73	68.25	63.66	58.02	53.76	61.32
BERT$_{base}$ (cls before pooler)	21.53	32.11	21.28	37.89	44.24	20.29	42.42	31.39
BERT$_{base}$ (first-last avg.)	39.69	59.37	49.67	66.03	66.19	53.88	62.06	56.70
BERT$_{base}$-flow$^{\diamondsuit}$	58.40	67.10	60.85	75.16	71.22	68.66	64.47	66.55
BERT$_{base}$-whitening$^{\diamondsuit}$	57.83	66.90	60.90	75.08	71.31	68.24	63.73	66.28
IS-BERT$_{base}$$^{\heartsuit}$	56.77	69.24	61.21	75.23	70.16	69.21	64.25	66.58
CT-BERT$_{base}$$^{\diamondsuit}$	61.63	76.80	68.47	77.50	76.48	74.31	69.19	72.05
ConSERT$_{base}$$^{\spadesuit}$	64.64	78.49	69.07	79.72	75.95	73.97	67.31	72.74
SG-BERT$_{base}$$^{\bigstar}$	68.49	80.00	71.34	81.71	77.43	77.99	68.75	75.10
SimCSE-BERT$_{base}$ (unsup.)$^{\diamondsuit}$	68.40	82.41	**74.38**	80.91	**78.56**	76.85	72.23	76.25
AdCSE-BERT$_{base}$ (Ours)	**70.52**	**84.10**	74.18	**82.15**	78.42	**78.32**	**73.16**	**77.26**

Implementation Details. For pure BERT, we adopted model weights released by Huggingface's Transformers[3] for evaluation. `[CLS]` output from BERT (cls before pooler) and the average embedding of the first and last layers (first-last avg.) are reported in this paper. For GloVe embeddings, we used averaging word vectors as sentence embeddings and report the result from Reimers *et al.* [25]. For BERT-flow, BERT-whitening and CT-BERT, we report the results reproduced by Gao *et al.* [12] which share the same evaluation setting with us. For IS-BERT and ConSERT, we report the results under unsupervised settings from their original paper (Zhang *et al.* [30], Yan *et al.* [29]). In addition, results evaluated by model named Contrastive (BT + SG-OPT) in Kim *et al.* [17] are reported as our baseline for SG-BERT.

Our implementation is based on SimCSE (Gao *et al.* [12]) and AdCo (Hu *et al.* [15])[4]. For AdCSE reported here, the max sequence length is set to 32 and dropout rate of both encoder q and encoder p are set to 0.1 just like the BERT defaults. We set learning rates for encoder q and negative adversaries to 3e–5 and 3e–3 respectively. Momentums of encoder p and negative adversaries are set to 0.995 and 0.9 respectively. The temperature τ of NT-Xent loss is set to 0.05. Besides, both batch size and the number of negatives are set to 64. We removed the projection layer in the evaluation phase to make the model more generalizing. Following SimCSE (Gao *et al.* [12]), we evaluated on the development set of STS-B every 125 steps during training and save the best model checkpoint for testing.

[3] https://github.com/huggingface/transformers.

[4] Our code is publicly available at https://github.com/lirenhao1997/AdCSE.

4.2 Main Results

Evaluation results on 7 STS tasks of AdCSE and other baselines are presented in Table 1, where the best results are in bold. We have the following observations:

- AdCSE yielded the best performance on most of the STS tasks. Specifically, it outperformed the previous state-of-the-art models on STS12, STS13, STS15, STSB, and SICK-R tasks, while having a small gap to them on STS14 and STS16. Taking STS12 as an example, AdCSE improved over the strongest baseline by 3.1%. Overall, AdCSE improved the previous best-averaged spearman correlation from 76.25% to 77.26%. This verifies the significance of untied dual-encoder networks and negative adversaries.
- In addition, compared with pure BERT, methods introducing contrastive learning performed better on 7 STS tasks. We attribute the collapse of pure BERT to their limitations on sentence embedding. The introduction of comparative learning can alleviate this collapse of pure BERT.

4.3 Ablation Study

To get deep insight into AdCSE and verify the validity of its two components separately, we conducted an ablation study of AdCSE.

- w/o negative adversaries: We removed the negative adversaries in AdCSE and only kept the dual-encoder backbone network for embedding. In this case, only different in-batch samples were used as negatives for contrastive learning.
- w/o untied dual-encoder: the untied dual-encoder backbone network was replaced by two encoders with shared parameters while negative adversaries were kept in this model.
- w/o both: a model without neither negative adversaries nor the untied dual-encoder (instead, using two encoders with shared parameters as the backbone) were evaluated as the baseline.

Table 2. Ablation study on AdCSE.

Model	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
AdCSE	**70.52**	**84.10**	**74.18**	82.15	**78.42**	78.32	**73.16**	**77.26**
- w/o negative adversaries	69.51	82.13	73.34	**82.49**	78.32	**78.60**	72.00	76.63
- w/o untied dual-encoder	67.75	77.22	70.01	80.46	77.59	76.48	68.86	74.05
- w/o both	66.02	79.83	69.90	76.42	75.54	74.33	70.07	73.16

With other settings held constantly, evaluation results of the ablated models on test set of STS tasks are shown in Table 2. According to the results, we observe that the best results of most evaluation tasks were obtained by complete AdCSE.

Removal of the untied dual-encoder and the negative adversaries separately led to a decrease in model performance, which indicates the contribution of both parts of AdCSE. It is worth noting that, the negative adversaries had about the same improvement in model performance with shared-parameter encoders (73.16 → 74.05) and the untied dural-encoder (76.63 → 77.26) as backbone, respectively, which verified the stability of adversarial training in AdCSE.

5 Further Analysis

We further validate AdCSE by analyzing sentence representations as well as the real cases in semantic similarity comparison. Impacts of both batch size and temperature are also investigated in this section.

5.1 Analysis of Embedding Space

To evaluate the quality of the embedding space of the model, we employed two metrics proposed by Wang *et el.* [28] for contrastive learning which called *alignment* and *uniformity*. For the embedding process $\mathcal{F}$, alignment ℓ_{align} is defined with the expected distance between positive pairs, while uniformity ℓ_{uniform} is the logarithm of the average pairwise Gaussian potential:

$$\ell_{\text{align}}(\mathcal{F}) \triangleq \mathbb{E}_{(x,y)\sim p_{\text{pos}}} \left[\|\mathcal{F}(x) - \mathcal{F}(y)\|_2^2\right] \tag{10}$$

$$\ell_{\text{uniform}}(\mathcal{F}) \triangleq \log \mathbb{E}_{x,y \overset{i.i.d}{\sim} p_{\text{data}}} \left[e^{-2\|\mathcal{F}(x)-\mathcal{F}(y)\|_2^2}\right] \tag{11}$$

where p_{pos} denotes the distribution of positive pairs and p_{data} denotes the data distribution. In Fig. 3, we showed alignment and uniformity of the sentence representations from some sentence embedding methods, where the averaged evaluation results on STS tasks were also reported along with the scatter points.

It can be seen that, embeddings from BERT encoders were better in alignment compared to the contrastive-based methods SimCSE and AdCSE, while their uniformity was worse which is the main reason for their poor evaluation results on STS tasks. The uniformity of AdCSE was slightly worse than that of unsupervised SimCSE, while it had a better alignment. In general, the contrastive learning effectively improves uniformity of pre-trained embeddings whereas keeping a good alignment. Moreover, the addition of negative adversaries further improves alignment, resulting in further model performance improvements.

5.2 Case Study on Semantic Similarity

Besides the performance evaluated by spearman correlation, the discrimination ability of models could be presented through a case study on semantic similarity calculation in an intuition way. Table 3 shows real cases from the development set of STS-B task where BERT, SimCSE and AdCSE were evaluated by measuring

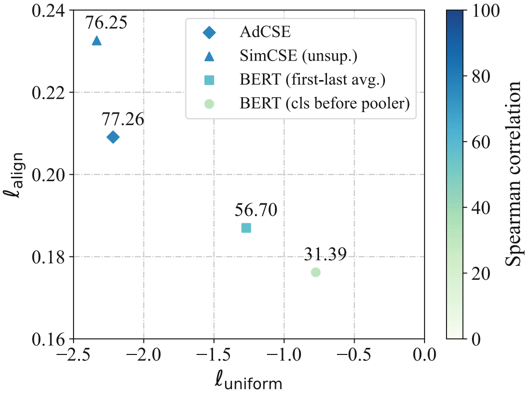


Fig. 3. Alignment and uniformity of some sentence embedding methods along with their averaged evaluation results on STS tasks. For both ℓ_{align} and ℓ_{uniform}, the lower the better.

Table 3. Case study on semantic similarity measuring, where GT is the ground truth similarity relatedness of the sentence pair, scored in [0, 5]. The predicted scores of models were obtained by mapping cosine similarities to the range of ground truth score. For each sentence pair, we bolded the predicted score closest to its ground truth.

	Sentence pair	GT	BERT (First-last avg.)	SimCSE (unsup.)	AdCSE
#1	A person drops a camera down an escelator. A man tosses a bag down an escalator.	2.75	3.75	3.30	**3.21**
#2	A woman is cutting some herbs. A woman is chopping cilantro.	2.80	4.22	3.49	**3.26**
#3	Five kittens are eating out of five dishes. Kittens are eating food on trays.	2.75	4.33	3.83	**3.10**

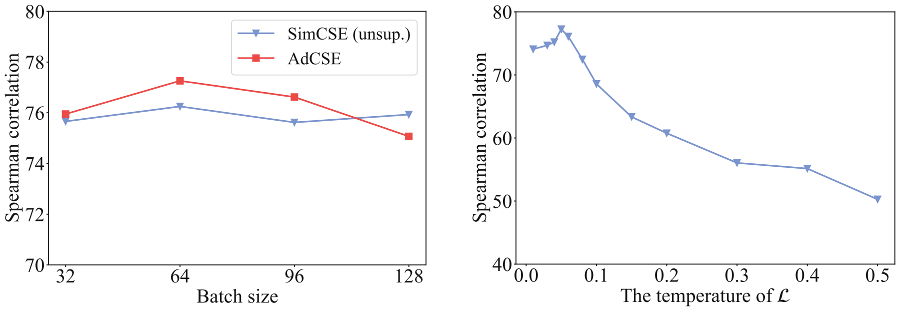
the cosine similarity of their output embeddings. Accroding to the results, BERT was far from the ground truth while SimCSE and AdCSE were able to measure the relatedness of sentence pairs more precisely. When it came to the hard case, SimCSE failed to handle the sentence pairs with high word coverage, such as (*Five kittens are eating out of five dishes*, *Kittens are eating food on trays*). In contrast, AdCSE could deal with this situation better thanks to the adversarial training with respect to hard negatives.

5.3 Influence of Batch Size and Temperature

We investigated the impact of batch size for training and temperature of $\mathcal{L}$ on model performance. Where we reported the averaged spearman correlation of the test set of all STS tasks as the model performance.

Batch Size. In some previous works of contrastive learning [9], a larger batch size may result in better performance. Thus, we experimented with different batch sizes of AdCSE on STS tasks. Note that we adjusted the number of negative adversaries accroding to the batch size in these experiments. As shown in Fig. 4(a), AdCSE benefited more from smaller batch sizes (32, 64, 96) compared to SimCSE, and achieved its best performance when the batch size was set to 64. A possible reason for this phenomenon is that a larger batch size together with more adversaries are in need of adjusting the corresponding learning rates α_q and α_n, which are hard to control for adversarial training.

Temperature. The hyperparameter temperature τ in Eq. 5 is used to control the smoothness of the distribution normalized by softmax operation. The distribution is smoothed by a large temperature while sharpened by a small one. Thus, an appropriate temperature can help the model learn from hard negatives by influencing its gradients during backpropagation. In our experiments, we explored the influence of temperature to AdCSE. As Fig. 4(b) shows, the best performance of AdCSE was reached with $\tau = 0.05$. Either too small or too large temperature affected the model's ability to learn from negative samples.



(a) Influence of different batch sizes

(b) Influence of different temperatures

Fig. 4. Performance analysis of batch size and temperature on STS tasks.

6 Conclusion

In this paper, we design a novel unsupervised sentence embedding method, named AdCSE, which consists of an untied dual-encoder backbone network and a group of negative adversaries. Employing contrastive learning as an objective, AdCSE is able to learn expressive sentence representations by adversarial training. Evaluation results on 7 STS tasks indicate that AdCSE is competitive compared with state-of-the-art methods. With ablation empirical evidence and in-depth analysis, we show the importance of each part of AdCSE and validate its effectiveness from different perspectives.

In the future, we plan to take advantage of information from different layers in BERT to improve the performance of AdCSE.

References

1. Agirre, E., et al.: SemEval-2015 Task 2: semantic textual similarity, English, Spanish and pilot on interpretability. In: SemEval@NAACL-HLT, pp. 252–263 (2015)
2. Agirre, E., et al.: SemEval-2014 Task 10: multilingual semantic textual similarity. In: SemEval@COLING, pp. 81–91 (2014)
3. Agirre, E., et al.: SemEval-2016 Task 1: semantic textual similarity, monolingual and cross-lingual evaluation. In: SemEval@NAACL-HLT, pp. 497–511 (2016)
4. Agirre, E., Cer, D.M., Diab, M.T., Gonzalez-Agirre, A.: SemEval-2012 Task 6: a pilot on semantic textual similarity. In: SemEval@NAACL-HLT, pp. 385–393 (2012)
5. Agirre, E., Cer, D.M., Diab, M.T., Gonzalez-Agirre, A., Guo, W.: *SEM 2013 shared task: semantic textual similarity. In: *SEM, pp. 32–43 (2013)
6. Carlsson, F., Gyllensten, A.C., Gogoulou, E., Hellqvist, E.Y., Sahlgren, M.: Semantic re-tuning with contrastive tension. In: ICLR (2021)
7. Cer, D.M., Diab, M.T., Agirre, E., Lopez-Gazpio, I., Specia, L.: SemEval-2017 Task 1: semantic textual similarity multilingual and crosslingual focused evaluation. In: SemEval@ACL, pp. 1–14 (2017)
8. Chen, Q., Zhu, X., Ling, Z., Wei, S., Jiang, H., Inkpen, D.: Enhanced LSTM for natural language inference. In: ACL, pp. 1657–1668 (2017)
9. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.E.: A simple framework for contrastive learning of visual representations. In: ICML, pp. 1597–1607 (2020)
10. Conneau, A., Kiela, D., Schwenk, H., Barrault, L., Bordes, A.: Supervised learning of universal sentence representations from natural language inference data. In: EMNLP, pp. 670–680 (2017)
11. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT, pp. 4171–4186 (2019)
12. Gao, T., Yao, X., Chen, D.: SimCSE: simple contrastive learning of sentence embeddings. In: EMNLP (2021)
13. Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality reduction by learning an invariant mapping. In: CVPR, pp. 1735–1742. IEEE Computer Society (2006)
14. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.B.: Momentum contrast for unsupervised visual representation learning. In: CVPR, pp. 9726–9735 (2020)
15. Hu, Q., Wang, X., Hu, W., Qi, G.: AdCo: adversarial contrast for efficient learning of unsupervised representations from self-trained negative adversaries. In: CVPR, pp. 1074–1083 (2021)
16. Kalchbrenner, N., Grefenstette, E., Blunsom, P.: A convolutional neural network for modelling sentences. In: ACL, pp. 655–665 (2014)
17. Kim, T., Yoo, K.M., Lee, S.: Self-guided contrastive learning for BERT sentence representations. In: ACL/IJCNLP, pp. 2528–2540 (2021)
18. Kiros, R., et al.: Skip-thought vectors. In: NeurIPS, pp. 3294–3302 (2015)
19. Li, B., Zhou, H., He, J., Wang, M., Yang, Y., Li, L.: On the sentence embeddings from pre-trained language models. In: EMNLP, pp. 9119–9130 (2020)
20. Logeswaran, L., Lee, H.: An efficient framework for learning sentence representations. In: ICLR (2018)

21. Marelli, M., Menini, S., Baroni, M., Bentivogli, L., Bernardi, R., Zamparelli, R.: A SICK cure for the evaluation of compositional distributional semantic models. In: LREC, pp. 216–223 (2014)
22. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: NeurIPS, pp. 3111–3119 (2013)
23. Palangi, H., et al.: Deep sentence embedding using long short-term memory networks: analysis and application to information retrieval. IEEE ACM Trans. Audio Speech Lang. Process. **24** (2016)
24. Pennington, J., Socher, R., Manning, C.D.: Glove: global vectors for word representation. In: EMNLP, pp. 1532–1543 (2014)
25. Reimers, N., Gurevych, I.: Sentence-BERT: sentence embeddings using Siamese BERT-networks. In: EMNLP-IJCNLP. pp. 3980–3990 (2019)
26. Reimers, N., Schiller, B., Beck, T., Daxenberger, J., Stab, C., Gurevych, I.: Classification and clustering of arguments with contextualized word embeddings. In: ACL, pp. 567–578 (2019)
27. Su, J., Cao, J., Liu, W., Ou, Y.: Whitening sentence representations for better semantics and faster retrieval. CoRR abs/2103.15316 (2021)
28. Wang, T., Isola, P.: Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In: ICML, pp. 9929–9939 (2020)
29. Yan, Y., Li, R., Wang, S., Zhang, F., Wu, W., Xu, W.: ConSERT: a contrastive framework for self-supervised sentence representation transfer. In: ACL/IJCNLP, pp. 5065–5075 (2021)
30. Zhang, Y., He, R., Liu, Z., Lim, K.H., Bing, L.: An unsupervised sentence embedding method by mutual information maximization. In: EMNLP, pp. 1601–1610 (2020)