Instance Smoothed Contrastive Learning for Unsupervised Sentence Embedding

Hongliang He[1,2*], Junlei Zhang[1,2*], Zhenzhong Lan[2,3†], Yue Zhang[2,3†]

[1]Zhejiang University, China
[2]School of Engineering, Westlake University, China
[3]Institute of Advanced Technology, Westlake Institute for Advanced Study, China
{hehongliang, zhangjunlei, lanzhenzhong, zhangyue}@westlake.edu.cn

Abstract

Contrastive learning-based methods, such as unsup-SimCSE, have achieved state-of-the-art (SOTA) performances in learning unsupervised sentence embeddings. However, in previous studies, each embedding used for contrastive learning only derived from one sentence instance, and we call these embeddings **instance-level** embeddings. In other words, each embedding is regarded as a unique class of its own, which may hurt the generalization performance. In this study, we propose IS-CSE (instance smoothing contrastive sentence embedding) to smooth the boundaries of embeddings in the feature space. Specifically, we retrieve embeddings from a dynamic memory buffer according to the semantic similarity to get a positive embedding group. Then embeddings in the group are aggregated by a self-attention operation to produce a **smoothed instance** embedding for further analysis. We evaluate our method on standard semantic text similarity (STS) tasks and achieve an average of 78.30%, 79.47%, 77.73%, and 79.42% Spearman's correlation on the base of BERT-base, BERT-large, RoBERTa-base, and RoBERTa-large respectively, a 2.05%, 1.06%, 1.16% and 0.52% improvement compared to unsup-SimCSE.

Introduction

Learning better universal sentence embedding (Gao, Yao, and Chen 2021) can benefit many natural language processing tasks, such as sentiment analysis, information retrieval and semantic search (Klein and Nabi 2022; Zhang et al. 2018; Pilehvar and Navigli 2015), and thus has received much attention. Recently, it has been shown that the contrastive learning-based methods give strong results for sentence embeddings (Gao, Yao, and Chen 2021; Wang et al. 2022; Zhou et al. 2022; Zhang et al. 2021). The core idea of contrastive learning is that positive and negative embedding pairs are generated given a batch of training sentences. Whereas the positive embeddings are often obtained via augmentation, and negative embeddings are sampled from a random collection of sentences. Following the construction of pairs, contrastive learning forces the model to learn discriminative embeddings by pulling positive sentence pairs together and pushing apart negative ones.

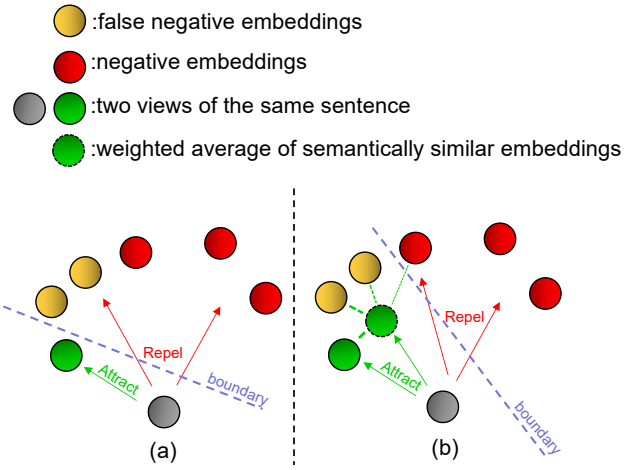

Figure 1: Comparison between our method with SimCSE. In SimCSE, two views of the same input sentence are regarded as positive pairs. Other sentences in the same batch are regarded as negative examples. (a): In SimCSE, each embedding is derived from one sentence, and one view of the input sentence is regarded as a label of another view. (b): Our method uses additional soft labels (weighted average of closing-by embeddings).

In the unsupervised contrastive learning framework, while some works seek to optimize for selecting "hard" negative examples (Zhou et al. 2022) or using pre-defined prompt (Jiang et al. 2022) to extract features, other methods investigate the effects of augmentation on constructing sentence pairs. One of the most influential methods for learning sentence embeddings is SimCSE (Gao, Yao, and Chen 2021), which takes drop-out as data augmentation, providing expressive semantically similar embeddings to construct positive pairs. ESimCSE(Wu et al. 2021) augmented the input sentences by word repetition, insertion, and deletion. Similarly, CARDS (Wang et al. 2022) randomly flip the first letter in a word to augment the inputs.

However, most of these methods take each of the sentences as a unique class and discriminate it from other sentences in a batch. This could make models become "overconfident" about each sentence being a separate class, be-

*These authors contributed equally.
†Corresponding authors.

cause there may be some false negative pairs in an unsupervised setting. To address this problem, DCLR (Zhou et al. 2022) generates negative examples by sampling them from a learned gaussian distribution and filtering out negative examples with high similarities. However, DCLR does not make use of rich positive embeddings. Inspired by the success of label smoothing (Müller, Kornblith, and Hinton 2019) where soft labels are applied to release the "over-confident" of a network caused by hard labels, we propose to smooth the positive examples to release the "over-confident" problem. For the positive pairs in contrastive learning, one positive embedding can be regarded as a label which another positive one should fit. Following the label smoothing method, we smooth the label by a weighted average operation with retrieved semantically similar embeddings. Specifically, we hold a First-in-First-out memory buffer which saves the sentence embeddings in the previous steps during the training process. While constructing the positive pairs, we retrieve sentence embeddings from the memory buffer based on the cosine similarity and do a weighted average operation with the positive embedding to get smooth embeddings. This can push each sentence to be similar to other closing-by sentences, not just itself. This new practice has a label smoothing effect (Szegedy et al. 2016). We call it instance smoothing to contrast sentence embedding (IS-CSE).

We evaluate IS-CSE on seven standard semantic textual similarity (STS) tasks (Agirre et al. 2012, 2013, 2014, 2015, 2016; Cer et al. 2017; Marelli et al. 2014) and 7 transfer learning tasks (Pang and Lee 2005; Hu and Liu 2004; Wiebe, Wilson, and Cardie 2005; Socher et al. 2013; Voorhees and Tice 2000; Dolan and Brockett 2005). Results show that our unsupervised model achieves a 79.47% and 79.42% averaged Spearman's correlation respectively using BERT$_{large}$ and RoBERTa$_{large}$, significantly outperforming competitive baselines on STS tasks. To better understand the effect of group-level embedding, we also calculate the alignment score (Wang and Isola 2020) between semantically similar positive pairs and the uniformity score of the whole representation to measure the quality of learned embeddings. We find that IS-CSE achieves better alignment results. But there is a little drop in uniformity except for the BERT$_{large}$ model. To the best of our knowledge, IS-CSE is the first attempt to create positive pairs from a group of similar sentences rather than each sentence in contrastive learning of unsupervised sentence representations. Our code is available at https://github.com/dll-wu/IS-CSE

Related Work

Unsupervised Sentence Embedding Learning

Unsup-SimCSE (Gao, Yao, and Chen 2021) proposes a contrastive learning framework to finetune pre-trained BERT (Devlin et al. 2018) and RoBERTa (Liu et al. 2019), and significantly outperforms previous results. SimCSE is further enhanced by several follow-up studies from different prospects. Instead of simply representing the sentence with [CLS] token or averaged embeddings, PromptBERT (Jiang et al. 2022) uses prompt tokens to represent a sentence. Data augmentation methods (Wu et al. 2021; Yan et al. 2021) are also applied to produce more high-quality training samples. For example, ESimCSE (Wu et al. 2021) enhances the input sentences with a repetition operation; ConSERT (Yan et al. 2021) takes multiple data augmentation strategies to further generate views for contrastive learning. Besides data augmentation methods, DCLR (Zhou et al. 2022) proposes an instance weighting method to punish false negatives and generate noise-based negatives to guarantee the uniformity of the representation space. However, no previous work in this task has tried to smooth the positive instances with sampled semantically similar sentence embeddings. Our work mainly differs from previous unsupervised embedding learning methods in three prospects: 1) we use a dynamic buffer to reduce the computational consumption; 2) we aggregate the retrieved embeddings to form a smoothed embedding instead of using their instance-level embedding directly in previous works; 3) we use both the instance and smoothed instance embeddings for discrimination.

Contrastive Learning

Contrastive learning has been originated applied in computer vision (He et al. 2020; Chen et al. 2020) and information retrieval (Bian et al. 2021) and achieved significant performance improvement. Data augmentation strategies such as image rotation and random cropping (Gao, Yao, and Chen 2021; Bian et al. 2021; Li et al. 2020b) are used to produce augmented images. The augmented images are then used as positive images for discrimination, while other images in the same mini-batch are regarded as negative ones. For unsupervised sentence representation learning, SimCSE (Gao, Yao, and Chen 2021) adopts dropout as the data augmentation, which improves the results on semantic textual similarities tasks by a large margin. Subsequent studies further adopt token shuffling (Yan et al. 2021) and back translation (Fang et al. 2020) to augment positive examples for sentence representation learning. However, to the best of our knowledge, how to augment embeddings from the smoothing view has not been studied. We fill the gap by investigating the effect of embedding smoothing for unsupervised sentence embedding learning.

Method

Baseline

SimCSE (Gao, Yao, and Chen 2021) applies contrastive learning on the universal sentence learning problem, where instance-level sentence embeddings are used as the input of the InfoNCE loss (Oord, Li, and Vinyals 2018). Specifically, given a collection of input sentences $\{x_i\}_{i=1}^m$. SimCSE simply uses identical sentences to build the sentence pair, i.e. $x_i = x_i^+$, and feeds x_i into a Transformer encoder f_θ twice. Since independently sampled dropout masks are applied to fully-connected layers and attention probabilities in f_θ, two separate sentence embeddings h_i and h_i^+ are obtained. For a mini-batch of N samples, the training loss for unsupervised SimCSE (unsup-SimCSE):

$$\mathcal{L}_{instance} = -log \frac{e^{sim(h_i, h_i^+)/\tau}}{\sum_{j=1}^{N} e^{sim(h_i, h_j^+)/\tau}}, \quad (1)$$

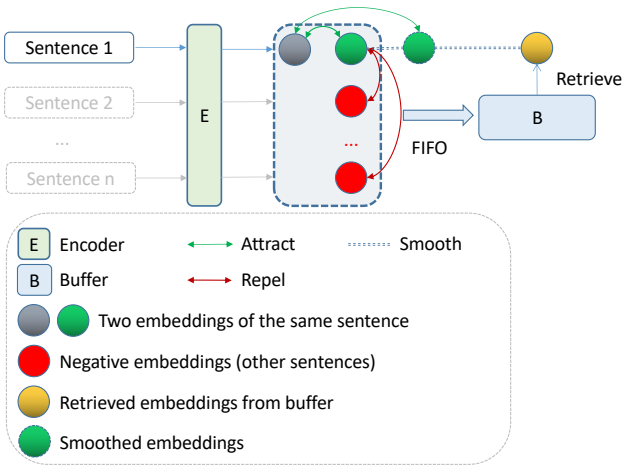

Figure 2: Overview of our method. We retrieve embeddings from the memory buffer (orange) and the smoothed embeddings are the weighted average of the retrieved and positive embeddings.

where τ is a temperature parameter and the $sim(\cdot,\cdot)$ represents the cosine similarity function:

$$sim(h_i, h_i^+) = \frac{h_i^T h_i^+}{\|h_i\| \, \|h_i^+\|}. \quad (2)$$

All embeddings h in Equ.1 are instance-level embeddings, each of which is derived from one sentence instance. In this paper, we propose an instance-smoothing mechanism to regularize the InfoNCE loss by applying smoothed instance embeddings (derived from a group of semantically similar sentences).

Dynamic Memory Buffer

Instead of only using embeddings derived from input sentences, we construct smoothed embeddings by averaging the closing-by embeddings. One key process of IS-CSE is to retrieve these closing-by embeddings at each step during fine-tuning. Directly retrieving sentence embeddings from the whole dataset can lead to a huge computational burden. To bound the memory usage, we propose to use a dynamic memory buffer in the unsupervised contrastive learning task. Specifically, given a dynamic memory buffer $\mathcal{B} \in \mathbb{R}^{L \times d}$, where L is the length of the buffer and d is the dimension of an embedding. For each step, we feed the buffer normalized augmented embeddings h^+ with a First-in-first-out (FIFO) strategy. The embeddings in memory buffer $\mathcal{B}$ are stop-gradient embeddings. Formally, the method for updating the memory buffer $\mathcal{B}$ is:

$$\mathcal{B}_{new} = Concat(\mathcal{B}_{old}[l:L], sg\{\frac{h_1^+}{||h_1^+||}, ..., \frac{h_l^+}{||h_l^+||}\}), \quad (3)$$

where l is the number of coming/discarded embeddings for the FIFO strategy (l equals the batch size in our experiment), sg is the stop-gradient operation and $Concat$ operation is used to maintain the buffer size and dynamically update the buffer. Based on the memory buffer, several semantically similar embeddings are retrieved for smoothing the augmented positive embedding h^+.

Retrieving Sentence Embeddings

After setting up the dynamic memory buffer, we retrieve sentence representations and apply the weighted average operation to get the smoothed embeddings. We compare two types of retrieval methods: kNN and K-means.

kNN A simple way to obtain semantically similar embeddings is kNN (Peterson 2009). Given the augmented embedding h^+, we calculate the cosine similarly (Equ.2) between h^+ and each of the embedding in buffer $\mathcal{B}$. Then k nearest embeddings are retrieved from $\mathcal{B}$.

K-means We perform the K-means algorithm (Hartigan and Wong 1979) on $\mathcal{B}$ with a pre-defined number of clusters k'. We assign each embedding to a cluster based on semantic similarity. We directly retrieve the center embedding to which h^+ belongs.

We empirically compare the performances of kNN and K-means in Table 3 and select the kNN as our final retrieval method.

Smoothing Instance Embeddings

In IS-CSE, the augmented embeddings h^+ are smoothed by retrieved embeddings with high semantic similarity from the dynamic buffer. For kNN, we apply a self-attention aggregation method. Specifically, given k retrieved embeddings $\{h^r\}_{i=1}^k$ and the augmented embedding h^+, we normalized and then concatenate them to get a combined matrix $K = \{h^+, h_1^r, h_2^r, ..., h_k^r\} \in \mathbb{R}^{(k+1)\times d}$. We thus obtain smoothed embedding h^{s+} by:

$$h^{s+} = softmax(\frac{h^+ K^T}{\beta})K, \quad (4)$$

where β is a temperature parameter.

For K-means, we cluster all the embeddings in the buffer based on the cosine similarity. Then we obtain a list of cluster centers, and select the center c^+ of the cluster which h^+ belongs to. We get our smoothed embedding h^{s+} by:

$$h^{s+} = \gamma h^+ + (1-\gamma)c^+, \quad (5)$$

where γ is a hyper-parameter. In Equ.4 and Equ.5, h^+ is not a stop-gradient embedding but the retrieved embeddings h^r and centers c^+ are stop-gradient embeddings.

Instance Smoothing Contrastive Sentence Embedding (IS-CSE)

The main difference between our method and SimCSE is that we add an additional contrastive loss whose augmented positive embeddings are smoothed. Given a batch of input sentences, we obtain the projected instance-level embeddings of h_i and h_i^+. We calculate our smoothed embedding

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
BERT$_{base}$(first-last avg.)*	39.70	59.38	49.67	66.03	66.19	53.87	62.06	56.80
BERT$_{base}$-flow*	58.40	67.10	60.85	75.16	71.22	68.66	64.47	66.55
BERT$_{base}$-whitening*	57.83	66.90	60.90	75.08	71.31	68.24	63.74	66.28
IS-BERT$^{*}_{base}$	56.77	69.24	61.21	75.23	70.16	69.21	64.25	66.58
CT-BERT$^{*}_{base}$	61.63	76.80	68.47	77.50	76.48	74.31	69.19	72.05
SimCSE-BERT$^{*}_{base}$	68.40	82.41	74.38	80.91	78.56	76.85	72.23	76.25
DLCR-BERT$^{*}_{base}$	70.81	83.73	75.11	82.56	78.44	78.31	71.59	77.22
IS-CSE-BERT$_{base}$	**72.86**	**84.02**	**76.35**	**82.64**	**78.65**	**79.53**	**74.05**	**78.30**
SimCSE-BERT$^{*}_{large}$	70.88	84.16	76.43	84.50	79.76	79.26	73.88	78.41
DCLR-BERT$_{large}$	71.87	84.83	77.37	84.70	**79.81**	79.55	74.19	78.90
IS-CSE-BERT$_{large}$	**73.76**	**85.06**	**78.14**	**85.02**	79.59	**80.43**	**74.30**	**79.47**
RoBERTa$_{base}$(fist-last avg.)*	40.88	58.74	49.07	65.63	61.48	58.55	61.63	56.57
RoBERTa$_{base}$-whitening*	46.99	63.24	57.23	71.36	68.99	61.36	62.91	61.73
DeCLUTR-RoBERTa$^{*}_{base}$	52.41	75.19	65.52	77.12	78.63	72.41	68.62	69.99
SimCSE-RoBERTa$^{*}_{base}$	70.16	81.77	73.24	81.36	80.65	80.22	68.56	76.57
DCLR-RoBERTa$_{base}$	70.01	**83.08**	**75.09**	**83.66**	81.06	**81.86**	**70.33**	**77.87**
IS-CSE-RoBERTa$_{base}$	**71.39**	82.58	74.36	82.75	**81.61**	81.40	69.99	77.73
SimCSE-RoBERTa$^{*}_{large}$	72.86	83.99	75.62	84.77	81.80	81.98	71.26	78.90
DCLR-RoBERTa$^{\dagger}_{large}$	73.09	84.57	76.13	85.15	81.99	82.35	71.80	79.30
DCLR-RoBERTa$_{large}$ (ours)	71.30	84.67	76.17	84.65	81.62	81.93	72.29	78.95
CARDS-RoBERT$_{large}$ (ours)	**74.78**	86.42	79.02	85.95	82.36	83.65	70.81	80.46
IS-CSE-RoBERTa$_{large}$	72.84	85.02	76.99	85.58	80.93	82.87	71.68	79.42
+ DCLR	73.67	85.46	76.86	85.16	81.31	82.25	71.71	79.49
+ CARDS	74.30	**86.47**	**79.06**	**85.99**	**82.78**	**84.02**	**72.80**	**80.77**

Table 1: Sentence embedding performance on STS tasks (Spearman's correlation). The best performance and the second-best performance with the same pre-trained encoder are denoted in bold and underlined fonts respectively. *: results from (Gao, Yao, and Chen 2021); †: results from (Zhou et al. 2022); (ours): our reproduced results based on code released by their authors; We add our $L_{smoothing}$ to the DCLR to get combined results and show it on "+DCLR". All the experiments are conducted in an unsupervised setting.

h_i^{s+} using Equ.4. The smoothed embedding loss can be calculated by:

$$\mathcal{L}_{smoothing} = -log\frac{e^{sim(h_i,h_i^{s+})/\tau}}{\sum_{j=1}^{N} e^{sim(h_i,h_j^{s+})/\tau}}. \quad (6)$$

Combining Equ.1 and Equ.6, we treat the smoothing loss as a regularizer. The final form of our training objective is:

$$\mathcal{L} = \mathcal{L}_{instance} + \alpha\mathcal{L}_{smoothing}, \quad (7)$$

where α is a coefficient.

The quality of retrieved embeddings may be low at the initial stages because the model has not been fully finetuned. A big α may hurt the model performance at the initial stages of finetuning. We adopt a cosine scheduler for α:

$$\alpha = \min\{\cos(\pi \cdot \frac{T_i}{T_{max}}) * (\alpha_{start} - \alpha_{end}), 0\} + \alpha_{end}, \quad (8)$$

where α_{start} , α_{end}, T_i and T_{max} are the inital value of α, end value of α, the current step and the max step, respectively.

Experiments

Setup

For unsupervised sentence embedding learning, we follow the same training process as SimCSE (Gao, Yao, and Chen 2021). We conduct our main experiments on 7 standard semantic textual similarities (STS) tasks: STS 2012-2016 (Agirre et al. 2012, 2013, 2014, 2015, 2016), STS Benchmark (Cer et al. 2017) and SICK-Relatedness (Marelli et al. 2014). We compare our IS-CSE against methods reported in SimCSE (Gao, Yao, and Chen 2021) and SimCSE-related methods: DCLR (Zhou et al. 2022), CARD (Wang et al. 2022). Although our method does not perform as good as CARD (Wang et al. 2022), we argue that CARD is an orthogonal method in that it finetunes BERT/RoBERTa with the help of finetuned models and additional data augmentation method, and can be combined with IS-CSE. We also include 7 transfer learning tasks (Conneau et al. 2017), taking STS as the main result for comparison following previous SimCSE-related papers (Gao, Yao, and Chen 2021; Wang et al. 2022; Zhou et al. 2022). Our experients are conducted on one NVIDIA A100 GPU.

	BERT		RoBERTa	
	base	large	base	large
Batch size	64	64	512	512
Learning rate	3e-5	1e-5	1e-5	3e-5

Table 2: Batch sizes and learning rates for IS-CSE

Group type	kNN	K-means	kNN+K-means
STS-B	**84.18**	83.74	84.14

Table 3: Results on STS-B development set of kNN group and K-means group using BERT$_{base}$ backbone. For the K-means group, the number of groups is 64 so the average number of embeddings in each group is equal to that in the kNN group. kNN+K-means denotes that two groups will be used and thus two smoothing objectives will be added. kNN retrieval method is finally selected.

Training Details

Our experimental settings are consistent with the SimCSE (Gao, Yao, and Chen 2021). Specifically, all our models are trained to start from the pre-trained checkpoints given by Huggingface (Wolf et al. 2020). Following SimCSE, the training corpus contains 10^6 sentences randomly sampled from English Wikipedia. We adopt [CLS] representation as the sentence embedding and an MLP pooler is used during training but discarded during inference. Hyperparameters for our model are the same as those for SimCSE. We train our model for 1 epoch and use the Adam optimizer (Kingma and Ba 2014). Cosine similarity with $\tau = 0.05$ is used to calculate sentence similarity. The details of batch size and learning rate are shown in Table 2. In IS-CSE, we set the buffer size $L = 1024$ and the number of kNN neighbors $k = 16$. According to the STS-B score on the development set in Table 3, we finally select the kNN group to apply our smoothing method. The temperature β for self-attention aggregation is set to 2. For BERT$_{base}$ and RoBERTa$_{base}$ we set $\alpha = 0.1$. For BERT$_{large}$ and RoBERTa$_{large}$ we set a cosine schedule (Equ. 8) for α from 0.005 to 0.05.

Main Results

We compare IS-CSE against previously published state-of-the-art unsupervised sentence embedding learning methods on STS tasks. We take the results reported in SimCSE for average GloVe embeddings (Pennington, Socher, and Manning 2014), average BERT or RoBERTa embeddings (Gao, Yao, and Chen 2021), BERT-flow (Li et al. 2020a), BERT-whitening (Su et al. 2021), unsup-SimCSE. For DCLR (Zhou et al. 2022), we take both the results reported on paper and our reproduced results based on their released code.

The results on 7 STS tasks are shown in Table 1. IS-CSE can outperform most previous competitive results on the basis of four different encoders (BERT$_{base}$, BERT$_{large}$, RoBERTa$_{base}$ and RoBERTa$_{large}$). Although we do not perform as well as DCLR on some of the tasks, IS-CSE is an orthogonal method in that it finetunes models with instance weighting, and may be combined with our methods. To evaluate it, we reproduce DCLR based on their released code and strictly follow their training settings. We further adding $L_{smoothing}$ to DCLR and the results ("+DCLR" in Table 1) indicate that IS-CSE can improve DCLR on most STS tasks.

Buffer size	STS-B	Avg. STS
256	82.75	78.95
512	83.85	79.48
1024	**84.18**	**79.91**
1536	83.02	78.35
2048	83.27	78.59
3072	83.60	78.92

Table 4: STS-B / Avg. STS development results with different buffer sizes using IS-CSE-BERT$_{base}$.

$N_{neighbors}$	8	12	16	20	24
STS-B	83.18	83.31	**84.18**	82.97	82.63

Table 5: Ablation studies of the number of neighbors on the STS-B development set using IS-CSE-BERT$_{base}$.

Ablation Studies

We investigate the impact of buffer size L, the hyperparameter α, β, and the number of neighbors in a group. All reported results in this section are based on the STS-B development set.

Buffer Size Table 4 shows the results of IS-CSE-BERT$_{base}$ with different buffer sizes. As can be seen from Table 4, when L increases from 256 to 1024, the performance also improves, which shows that larger buffers can allow more similar instances to be retrieved. However, a large buffer size beyond 1024 may cause performance degradation, this can be because a large buffer stores embeddings of several batches, and older embeddings are inconsistent with the current model parameters.

Number of Neighbors Table 5 shows the effects of different numbers of neighbors in kNN. We empirically find that IS-CSE performs well when $k = 16$, which is probably because that smoothing is not sufficient when k is smaller than 16 and some noise samples will be introduced when k is greater than 16.

Hyperparameter α In IS-CSE, α is used as the weight of the $L_{smoothing}$. We tried two types of α: constant α and dynamic α. For the former, we just assign a constant value to α and never change it during finetuning; For the latter, we use a cosine schedule function (Equ. 8) to gradually increase the value from α_{start} to α_{end}. Table 6 shows the result of applying different constant α and dynamic α on IS-CSE-RoBERTa$_{large}$. We empirically find that BERT$_{large}$ and RoBERTa$_{large}$ can perform better with dynamic α.

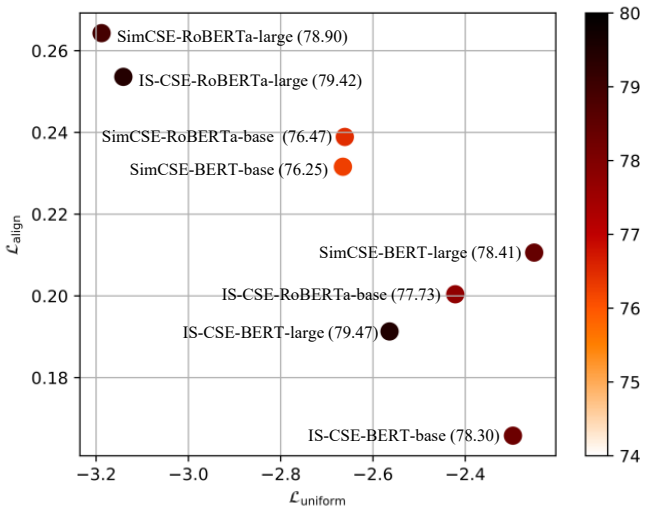

Figure 3: $\mathcal{L}_{align}$-$\mathcal{L}_{uniform}$ plot contains the alignment and uniformity measurements for our models. The color of points and numbers in brackets represent average STS performance.

	α	α_{start}	α_{end}	STS-B
Constant α	0.01	-	-	85.12
	0.05	-	-	84.42
	0.1	-	-	83.57
Dynamic α	-	0.005	0.05	**85.76**
	-	0.05	0.1	84.47
	-	0.005	0.1	84.99

Table 6: Effects of different α schedules on STS-B development set for IS-CSE-RoBERTa$_{large}$.

β	1	2	3	4
STS-B	83.78	**84.18**	83.07	81.58

Table 7: Comparison of different constant β on STS-B development set using IS-CSE-BERT$_{base}$.

Hyperparameter β After finishing the retrieval process, we perform self-attention aggregation on a group of embeddings to smooth the representation. In Table 7, we compare the impact of choosing different β on STS-B development set. β is used to adjust the attention weights, and a larger β will make the attention weights more even.

Analysis

In this section, we conduct further analyses to verify the effectiveness of IS-CSE.

Alignment and Uniformity Alignment and Uniformity are two key properties of embedding learned by contrastive loss (Wang and Isola 2020). It has been shown that models

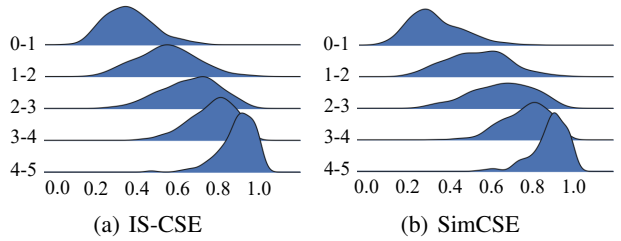

Figure 4: Density plots of cosine similarities between sentence pairs in STS-B based on the RoBERTa-large model. Sentence pairs are divided into 5 groups based on ground truth scores of similarity (higher means more similar) along the y-axis, and the x-axis is the cosine similarity.

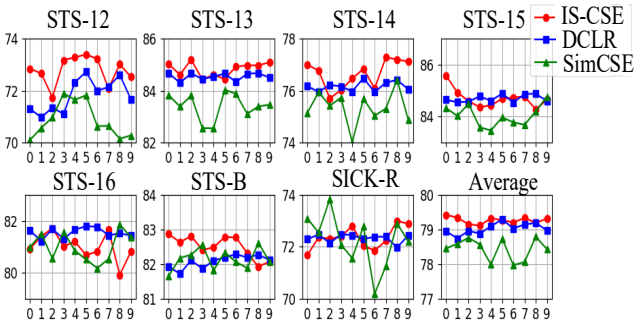

Figure 5: We compare our method on STS tasks with 10 random seeds based on RoBERTa-large model.

which have both better alignment and uniformity can perform better on sentence representation. Figure 3 shows the uniformity and alignment of different sentence embedding models along with their STS results. Our smoothing method IS-CSE achieves better alignment on all four backbones. However, compare to SimCSE, there are some adverse effects on the uniformity of base models. For large models, the uniformity only drops by 0.04 on RoBERTa$_{large}$ and increases by 0.31 on BERT$_{large}$. The results show that IS-CSE can achieve a better balance between alignment and uniformity.

Transfer Learning We evaluate our models on the 7 transfer learning tasks: MR (Pang and Lee 2005), CR (Hu and Liu 2004), MPQA (Wiebe, Wilson, and Cardie 2005), SST2 (Socher et al. 2013), TREC (Voorhees and Tice 2000) and MRPC (Dolan and Brockett 2005). We train a logistic regression classifier on top of frozen finetuned encoders produced by different methods. We follow the default configurations in (Gao, Yao, and Chen 2021).

The results on transfer tasks are shown in Table 8. IS-CSE achieves better or on par results than previous approaches except for BERT$_{base}$. This indicates that IS-CSE has better transferability than other models on most tasks.

Results with Different Seeds To evaluate the stability of our method, we show the test result with four different seeds in Figure 5. Our model can outperform SimCSE and DCLR in most seeds and tasks.

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-BERT$_{base}$	81.09	**87.18**	**94.96**	88.75	**85.96**	88.64	74.24	**85.83**
SimCSE-BERT$_{base}$	**81.18**	86.46	94.45	88.88	85.50	89.80	74.43	85.81
IS-CSE-BERT$_{base}$	80.48	85.32	94.67	**89.44**	85.06	87.40	**75.77**	85.45
SimCSE-RoBERTa$_{base}$	81.04	87.74	**93.28**	86.94	86.60	**84.60**	73.68	84.84
IS-CSE-RoBERTa$_{base}$	**81.93**	**87.76**	93.24	**87.61**	**87.48**	83.20	**76.35**	**85.37**
SimCSE-RoBERTa$_{large}$	**82.74**	**87.87**	**93.66**	88.22	88.58	92.00	69.68	86.11
IS-CSE-RoBERTa$_{large}$	82.70	87.79	93.30	**88.36**	**89.02**	**92.40**	**74.96**	**86.93**

Table 8: Transfer task results of different sentence embedding models (measured as accuracy). Results for comparison are reported in published paper SimCSE (Gao, Yao, and Chen 2021). We highlight the highest numbers among models with the same pre-trained encoder.

Query Sentence	This can probably be attributed to the intelligence-gathering of german civilians based in ireland during the 1930s.	
Retrieved Sentences	1	The "luftwaffe" carried out a number of air raids against the midlands and england in the middle part of 1942.
	2	During the world war ii, the area became an important station for anti-activities
	3	Many union members were jewish and were killed during world war ii.
Query Sentence	"ravenswood" may refer to	
Retrieved Sentences	1	"roanoke" may refer to
	2	"yasir ali" may refer to
	3	"datuna" may refer to

Table 9: We show the retrieved sentences in our method. "Query Sentence" represents the sentence used as a query. "Retrieved Sentences" represents the sentence retrieved from the dynamic memory buffer.

Cosine-Similarity Distribution To directly evaluate our approaches to STS tasks, we illustrate the cosine similarity distribution of sentence pairs in the STS-B dataset with different groups of human ratings in Figure 4. Compared with SimCSE, our method has a more scattered distribution with lower variance and has a similar discrimination ability. This observation validates that our method can achieve a better alignment-uniformity balance.

Case Study of Retrieved Sentences We smooth the instance-level embeddings by aggregating retrieved embeddings. To better understand the smoothing process, we list the top three highest retrieved sentences based on kNN in Table 9. The "Query Sentence" is used as the query embedding during retrieval and the "Retrieved Sentences" are the top three highest sentences retrieved from the dynamic memory buffer according to the similarity. Though the meaning of retrieved sentences and the query sentences is not totally the same, they are similar semantically in some text segments. For example, the query sentence "ravenswood may refer to" has the same structure as retrieved sentence "roanoke" may refer to". Thus the retrieved sentences help to smooth the query sentence and achieve better performance on STS tasks.

Conclusion

We proposed IS-CSE, an instance smoothing contrastive learning framework for unsupervised sentence representation learning. Our main idea is to improve the generalization ability by smoothing the positive examples. Specifically, in our framework, we aggregate retrieved semantically similar instances from a dynamic memory buffer to produce group-level positive embeddings, which are then used for discrimination. Experimental results on seven STS tasks have shown that our approach outperforms several competitive baselines. Our instance-level smoothing method is general and can be applied to other settings in Contrastive Learning.

In the future, we will explore more granularities for smoothing positive sentences for discrimination. Whether negative examples can be smoothed will also be studied. We will also consider applying our method for more natural language processing tasks, such as summarization.

Acknowledgements

This research has been supported by the Key R&D program of Zhejiang Province (Grant No. 2021C03139). We also would like to thank Westlake University HPC Center for providing HPC support.

References

Agirre, E.; Banea, C.; Cardie, C.; Cer, D.; Diab, M.; Gonzalez-Agirre, A.; Guo, W.; Lopez-Gazpio, I.; Maritxalar, M.; Mihalcea, R.; 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)*, 252–263.

Agirre, E.; Banea, C.; Cardie, C.; Cer, D.; Diab, M.; Gonzalez-Agirre, A.; Guo, W.; Mihalcea, R.; Rigau, G.; and Wiebe, J. 2014. Semeval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, 81–91.

Agirre, E.; Banea, C.; Cer, D.; Diab, M.; Gonzalez Agirre, A.; Mihalcea, R.; Rigau Claramunt, G.; and Wiebe, J. 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).

Agirre, E.; Cer, D.; Diab, M.; and Gonzalez-Agirre, A. 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)*, 385–393.

Agirre, E.; Cer, D.; Diab, M.; Gonzalez-Agirre, A.; and Guo, W. 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*, 32–43.

Bian, S.; Zhao, W. X.; Zhou, K.; Cai, J.; He, Y.; Yin, C.; and Wen, J.-R. 2021. Contrastive curriculum learning for sequential user behavior modeling via data augmentation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 3737–3746.

Cer, D.; Diab, M.; Agirre, E.; Lopez-Gazpio, I.; and Specia, L. 2017. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*.

Chen, X.; Fan, H.; Girshick, R.; and He, K. 2020. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*.

Conneau, A.; Kiela, D.; Schwenk, H.; Barrault, L.; and Bordes, A. 2017. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dolan, B.; and Brockett, C. 2005. Automatically constructing a corpus of sentential paraphrases. In *Third International Workshop on Paraphrasing (IWP2005)*.

Fang, H.; Wang, S.; Zhou, M.; Ding, J.; and Xie, P. 2020. Cert: Contrastive self-supervised learning for language understanding. *arXiv preprint arXiv:2005.12766*.

Gao, T.; Yao, X.; and Chen, D. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *EMNLP (1)*.

Hartigan, J. A.; and Wong, M. A. 1979. Algorithm AS 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)*, 28(1): 100–108.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9729–9738.

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

Jiang, T.; Huang, S.; Zhang, Z.; Wang, D.; Zhuang, F.; Wei, F.; Huang, H.; Zhang, L.; and Zhang, Q. 2022. PromptBERT: Improving BERT Sentence Embeddings with Prompts. *arXiv preprint arXiv:2201.04337*.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Klein, T.; and Nabi, M. 2022. miCSE: Mutual Information Contrastive Learning for Low-shot Sentence Embeddings. *arXiv preprint arXiv:2211.04928*.

Li, B.; Zhou, H.; He, J.; Wang, M.; Yang, Y.; and Li, L. 2020a. On the sentence embeddings from pre-trained language models. *arXiv preprint arXiv:2011.05864*.

Li, G.; Zhang, J.; Wang, Y.; Liu, C.; Tan, M.; Lin, Y.; Zhang, W.; Feng, J.; and Zhang, T. 2020b. Residual distillation: Towards portable deep neural networks without shortcuts. *Advances in Neural Information Processing Systems*, 33: 8935–8946.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Marelli, M.; Menini, S.; Baroni, M.; Bentivogli, L.; Bernardi, R.; and Zamparelli, R. 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)*, 216–223.

Müller, R.; Kornblith, S.; and Hinton, G. E. 2019. When does label smoothing help? *Advances in neural information processing systems*, 32.

Oord, A. v. d.; Li, Y.; and Vinyals, O. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

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

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

Peterson, L. E. 2009. K-nearest neighbor. *Scholarpedia*, 4(2): 1883.

Pilehvar, M. T.; and Navigli, R. 2015. From senses to texts: An all-in-one graph-based approach for measuring semantic similarity. *Artificial Intelligence*, 228: 95–128.

Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C. D.; Ng, A. Y.; and Potts, C. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, 1631–1642.

Su, J.; Cao, J.; Liu, W.; and Ou, Y. 2021. Whitening sentence representations for better semantics and faster retrieval. *arXiv preprint arXiv:2103.15316*.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2818–2826.

Voorhees, E. M.; and Tice, D. M. 2000. Building a question answering test collection. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, 200–207.

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

Wang, W.; Ge, L.; Zhang, J.; and Yang, C. 2022. Improving Contrastive Learning of Sentence Embeddings with Case-Augmented Positives and Retrieved Negatives. *arXiv preprint arXiv:2206.02457*.

Wiebe, J.; Wilson, T.; and Cardie, C. 2005. Annotating expressions of opinions and emotions in language. *Language resources and evaluation*, 39(2): 165–210.

Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, 38–45.

Wu, X.; Gao, C.; Zang, L.; Han, J.; Wang, Z.; and Hu, S. 2021. Esimcse: Enhanced sample building method for contrastive learning of unsupervised sentence embedding. *arXiv preprint arXiv:2109.04380*.

Yan, Y.; Li, R.; Wang, S.; Zhang, F.; Wu, W.; and Xu, W. 2021. Consert: A contrastive framework for self-supervised sentence representation transfer. *arXiv preprint arXiv:2105.11741*.

Zhang, J.; Yao, H.; He, J.; and Sun, X. 2018. Illustrate your travel notes: web-based story visualization. In *Proceedings of the 10th International Conference on Internet Multimedia Computing and Service*, 1–5.

Zhang, J.; et al. 2021. S-SimCSE: sampled sub-networks for contrastive learning of sentence embedding. *arXiv preprint arXiv:2111.11750*.

Zhou, K.; Zhang, B.; Zhao, W. X.; and Wen, J.-R. 2022. Debiased Contrastive Learning of Unsupervised Sentence Representations. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 6120–6130.