

WhitenedCSE: Whitening-based Contrastive Learning of Sentence Embeddings

Wenjie Zhuo¹ Yifan Sun² Xiaohan Wang¹ Linchao Zhu¹ Yi Yang^{1†}

¹Zhejiang University, Hangzhou, China

²Baidu Inc., Beijing, China

{12021057, yangyics}@zju.edu.cn

sunyf15@tsinghua.org.cn

{wxh1996111, zhulinchao7}@gmail.com

Abstract

This paper presents a whitening-based contrastive learning method for sentence embedding learning (WhitenedCSE), which combines contrastive learning with a novel shuffled group whitening. Generally, contrastive learning pulls distortions of a single sample (*i.e.*, positive samples) close and push negative samples far away, correspondingly facilitating the alignment and uniformity in the feature space. A popular alternative to the “pushing” operation is whitening the feature space, which scatters all the samples for uniformity. Since the whitening and the contrastive learning have large redundancy *w.r.t.* the uniformity, they are usually used separately and do not easily work together. For the first time, this paper integrates whitening into the contrastive learning scheme and facilitates two benefits. **1) Better uniformity.** We find that these two approaches are not totally redundant but actually have some complementarity due to different uniformity mechanism. **2) Better alignment.** We randomly divide the feature into multiple groups along the channel axis and perform whitening independently within each group. By shuffling the group division, we derive multiple distortions of a single sample and thus increase the positive sample diversity. Consequently, using multiple positive samples with enhanced diversity further improves contrastive learning due to better alignment. Extensive experiments on seven semantic textual similarity tasks show our method achieves consistent improvement over the contrastive learning baseline and sets new states of the art, *e.g.*, 78.78% (+2.53% based on BERT_{base}) Spearman correlation on STS tasks.¹

1 Introduction

This paper considers self-supervised sentence representation (embedding) learning. It is a fundamental task in language processing (NLP) and can

[†]Corresponding author.

¹Our code will be available at <https://github.com/SupstarZh/WhitenedCSE>.

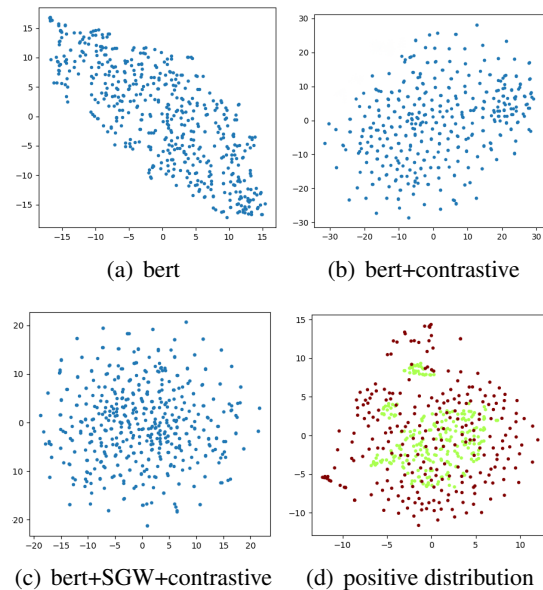


Figure 1: The uniformity gradually improves in the deep embedding of (a) bert, (b) bert + contrastive, and (c) bert + SGW + contrastive, *i.e.*, the proposed WhitenedCSE. Meanwhile, in (d), the positive samples after SGW (red) obtain higher diversity than the original bert features (green). Using these diverse positive samples for contrastive learning, the proposed WhitenedCSE achieves better alignment.

benefit a wide range of downstream tasks (Qiao et al., 2016; Le and Mikolov, 2014; Lan et al., 2019; Logeswaran and Lee, 2018). Two characteristics matter for sentence embeddings, *i.e.*, uniformity (of the overall feature distribution) and alignment (of the positive samples), according to a common sense in deep representation learning (Wang and Isola, 2020). Alignment expects minimal distance between positive pairs, while uniformity expects the features are uniformly distributed in the representation space in overall. From this viewpoint, the popular masked language modeling (MLM) (Devlin et al., 2018; Liu et al., 2019; Brown et al., 2020b; Reimers and Gurevych, 2019) is not an optimal choice for sentence embedding: MLM methods do

not explicitly enforce the objective of uniformity and alignment and thus do not quite fit the objective of sentence representation learning.

To improve the uniformity as well as the alignment, there are two popular approaches, *i.e.*, contrastive learning and post-processing. 1) The contrastive learning methods (Yan et al., 2021; Gao et al., 2021; Kim et al., 2021; Wang et al., 2021) pulls similar sentences close to each other and pushes dissimilar sentences far-away in the latent feature space. Pulling similar sentences close directly enforces alignment, while pushing dissimilar sentences apart implicitly enforces uniformity (Wang and Isola, 2020). 2) In contrast, the post-processing methods mainly focus on improving the uniformity. They use normalizing flows (Li et al., 2020) or whitening operation (Su et al., 2021)) to project the already-learned representations into an isotropic space. In other words, these methods scatter all the samples into the feature space and thus improve the uniformity.

In this paper, we propose a whitening-based contrastive learning method for sentence representation learning (WhitenedCSE). For the first time, we integrate whitening into the contrastive learning scheme and demonstrate substantial improvement. Specifically, WhitenedCSE combines contrastive learning with a novel Shuffled Group Whitening (SGW). Given a backbone feature, SGW randomly divides the feature into multiple groups along the channel axis and perform whitening independently within each group. The whitened features are then fed into the contrastive loss for optimization.

Although the canonical whitening (or group whitening) is only beneficial for uniformity, SGW in WhitenedCSE improves not only the uniformity but also the alignment. We explain these two benefits in details as below:

- Better uniformity. We notice that the pushing effect in contrastive learning and the scattering effect in the whitening have large redundancy to each other, because they both facilitate the uniformity. This redundancy is arguably the reason why no prior literature tries to combine them. Under this background, our finding *i.e.*, these two approaches are not totally redundant but actually have some complementarity is non-trivial. We think such complementarity is because these two approaches have different uniformity mechanism and will discuss the differences in Section 3.2.3. In Fig. 1, we observe that while the contrastive learning (Fig. 1 (b))

already improves the uniformity over the original bert features (Fig. 1 (a)), applying whitening (Fig. 1 (c)) brings another round of uniformity improvement.

- Better alignment. In the proposed WhitenedCSE, SGW is featured for its shuffled grouping operation, *i.e.*, randomly dividing a backbone feature into multiple groups before whitening. Therefore, given a same backbone feature, we may repeat SGW multiple times to get different grouping results, and then different whitened features. These “duplicated” features are different from each other and thus increase the diversity of positive samples, as shown in Fig. 1 (d). Using these diverse positive samples for contrastive learning, WhitenedCSE improves the alignment.

Another important advantage of SGW is: since it is applied onto the backbone features, it incurs very slight computational overhead for generating additional positive samples. This high efficiency allows WhitenedCSE to increase the number of positive samples (more than common setting of 2) in a mini-batch with little cost. Ablation study shows that the enlarged positive-sample number brings a further benefit.

Our contributions are summarized as follows:

- (1) We propose WhitenedCSE for the self-supervised sentence representation learning task. WhitenedCSE combines the contrastive learning with a novel Shuffled Group Whitening (SGW).

- (2) We show that through SGW, WhitenedCSE improves not only the uniformity but also the alignment. Moreover, SGW enables efficient multi-positive training, which is also beneficial.

- (3) We evaluate our method on seven semantic textual similarity tasks and seven transfer tasks. Experimental results show that WhitenedCSE brings consistent improvement over the contrastive learning baseline and sets new states of the art.

2 Related Work

2.1 Sentence Representation Learning

As a fundamental task in natural language processing, sentence representation learning has been extensively studied. Early works mainly based on bag-of-words (Wu et al., 2010; Tsai, 2012) or context prediction tasks (Kiros et al., 2015; Hill et al., 2016), *etc.* Recently, with the advent of pretrained language model (Devlin et al., 2018; Liu et al., 2019; Brown et al., 2020a), many works tend to directly use PLMs, such as BERT (Devlin et al.,

2018), to generate sentence representations. However, some studies (Ethayarajh, 2019; Yan et al., 2021) found that directly use the [CLS] representation or the average pooling of token embeddings at the last layer will suffer from *anisotropy* problem, *i.e.*, the learned embeddings are collapsed into a small area. To alleviate this problem, BERT-flow (Li et al., 2020) adopts a standardized flow transformation while BERT-Whitening (Su et al., 2021) adopts a whitening transformation, both of them transform the representation space to a smooth and isotropic space. Most recently, contrastive learning (Chen et al., 2020; Gao et al., 2021) has become a powerful tool to obtain the sentence representations.

2.2 Contrastive Learning

Contrastive learning (Chen et al., 2020; He et al., 2020) has achieved great success in sentence representation learning tasks (Gao et al., 2021; Yan et al., 2021; Kim et al., 2021; Wang et al., 2021). It pulls semantically similar samples together, and pushes the dissimilar samples away, which can be formulated as:

$$\mathcal{L}_{cl} = -\log \frac{e^{\text{sim}(h_i, h_i^*)/\tau}}{\sum_{j=1}^n e^{\text{sim}(h_i, h_j^*)/\tau}} \quad (1)$$

where τ is a temperature hyperparameter, h_i^* , h_j^* are the positive sample and negative samples respectively. Recently, *alignment* and *uniformity* (Wang and Isola, 2020) are proposed to measure the quality of representations. Alignment measures whether the distance between positive samples is close, while uniformity measures the dispersion of embedding in vector space. A typical method called SimCSE (Gao et al., 2021) uses dropout as a feature-wise data augmentation to construct the positive sample, and randomly sample negatives from the batch, which can achieve a great balance between alignment and uniformity. Some new works further improved the quality of sentence representations based on SimCSE, such as ESIMCSE (Wu et al., 2021), MixCSE (Zhang et al., 2022a) and VaSCL (Zhang et al., 2021b), each of them proposed a new data augmentation strategy to construct the positive pair. Besides, DCLR (Zhou et al., 2022) focus on optimizing the strategy of sampling negatives, and ArcCSE (Zhang et al., 2022b) optimized the objective function, *etc.* In this paper, we find that contrastive learning can be further combined with whitening to obtain better sentence representations.

2.3 Whitening Transformation

In computer vision, recent works (Ermolov et al., 2021; Zhang et al., 2021c; Hua et al., 2021) use whitening transformation as an alternative method to the "pushing negatives away" operation in contrastive learning to disperse the data uniformly throughout the spherical space (*i.e.*, the feature space), and then pull the positive samples together, which have achieved great success in unsupervised representation learning.

Whitening (*aka.*, sphering) is a common transformation that transforms a set of variables into a new set of isotropic variables, and makes the covariance matrix of whitened variables equal to the identity matrix. In natural language processing, Su et al. (2021) use whitening as a post-processing method to alleviate the anisotropic problem in pre-trained language models. In this paper, we use whitening as an explicit operation to further improve the uniformity of the representation space, and further explore the potential of whitening in improving alignment, so as to obtain a better sentence representation model.

3 Methods

In this section, we first describe the overall architecture of WhitenedCSE and then present the details of all the modules, including the shuffled group whitening module and the new contrastive learning module.

3.1 General Framework

As shown in Fig. 2, WhitenedCSE has three major components:

- An BERT-like encoder, which we use to extract features from native sentences, and take the [CLS] token as our native sentence representations.
- Shuffled-group-whitening module, we use it as a complementary module to contrastive learning to further improve the uniformity and alignment of the representation space.
- Multi-positives contrastive module, in this module, we pull distortions of the representations close and push the negative samples away in the latent feature space.

Specifically, given a batch of sentences \mathbf{X} , WhitenedCSE use the feature encoder $f_\theta(x_i, \gamma)$ to map them to a higher dimensional space, where γ is a random mask for dropout (Gao et al., 2021), then we take the [CLS] output as the native sentence representations. After this, we feed the na-

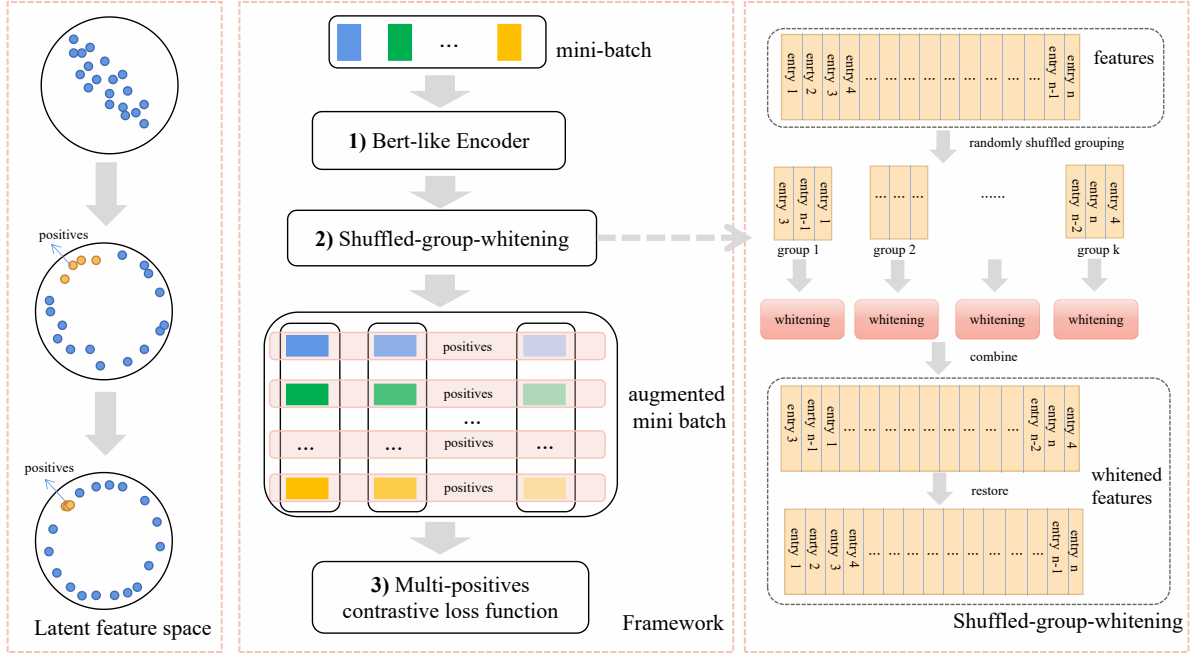


Figure 2: Method overview. In the middle column, Whitenecse consists of three components, *i.e.*, 1) An BERT-like encoder for generating the backbone features from input samples, 2) A Shuffled Group Whitenning (SGW) module for scattering the backbone features and augmenting the positive feature diversity, and 3) A multi-positive contrastive loss for optimizing the features. In the left column, when the mini-batch flows through these three components sequentially, the feature space undergoes “anisotropy” → “good uniformity + augmented positives” → “pulling close the positives”. The right column illustrates the SGW module in details. Specifically, SGW randomly shuffles the backbone feature along the axis and then divides the feature into multiple groups. Afterwards, SGW whitens each group independently and re-shuffles the whitened feature. Given a single backbone feature, we repeat the SGW process several times so as to generate multiple positive features.

tive sentence representations to the shuffled-group-whitening (SGW) module, in this module we randomly dividing each sentence representation into multiple groups along the axis, then we operate group whitening on each group. We repeat SGW multiple times to get different grouping results, and then different whitened representations. These "duplicated" features are different from each other. Finally, we use multi-positives contrastive loss function to pull one representation and all its corresponding augmentations together, and push it away from others. We will discuss feasible loss function in Section 3.3, and present our final form of loss function.

3.2 From Whitening to SGW

3.2.1 Preliminaries for Whitening

Given a batch of normalized sentence representations $\mathbf{Z} \in \mathbb{R}^{N \times d}$, the whitening transformation can be formulated as:

$$\mathbf{H} = \mathbf{Z}^T \mathbf{W} \quad (2)$$

where $\mathbf{H} \in \mathbb{R}^{d \times N}$ is the whitened embeddings

and $\mathbf{W} \in \mathbb{R}^{d \times d}$ is the whitening matrix. We denote the covariance matrix of $\mathbf{Z}\mathbf{Z}^T$ as Σ . the goal of whitening is to make the covariance matrix of $\mathbf{H}\mathbf{H}^T$ equal to the identity matrix \mathbf{I} , *i.e.*, $\mathbf{W}\Sigma\mathbf{W}^T = \mathbf{I}$. There are many different whitening methods, such as PCA (Jégou and Chum, 2012), ZCA (Bell and Sejnowski, 1997), etc. Group whitening use ZCA as its whitening method to prevent the *stochastic axis swapping* (Huang et al., 2018).²

ZCA Whitening. The whitening matrix of ZCA whitening transformation can be formulated as:

$$\mathbf{W}^{ZCA} = \mathbf{U}\mathbf{\Lambda}^{-1/2}\mathbf{U}^T \quad (3)$$

where $\mathbf{U} \in \mathbb{R}^{d \times d}$ is the stack of eigenvector of $cov(\mathbf{Z}, \mathbf{Z}^T)$, and $\mathbf{\Lambda}$ is the correspond eigenvalue matrix. \mathbf{U} and $\mathbf{\Lambda}$ are obtained by matrix decomposition. Therefore, Eq. 2 becomes:

$$\mathbf{H} = \mathbf{Z}^T \mathbf{U}\mathbf{\Lambda}^{-1/2}\mathbf{U}^T \quad (4)$$

²*stochastic axis swapping* can drastically change the data representation from one batch to another such that training never converges (Huang et al., 2018).

Group Whitening. Since whitening module needs a large batch size to obtain a suitable estimate for the full covariance matrix, while in NLP, large batch size can be detrimental to unsupervised contrastive learning. To address this problem, we use group whitening (Huang et al., 2018), which controls the extent of whitening by decorrelating smaller groups. Specifically, give a sentence representation of dimension d , group whitening first divide it into k groups ($\mathbf{Z}_0, \mathbf{Z}_1, \dots, \mathbf{Z}_{k-1}$), i.e., $\mathbf{Z}_k \in \mathbb{R}^{N \times \frac{d}{k}}$ and then apply whitening on each group. That is:

$$\mathbf{H} = \text{concat}(\mathbf{Z}_i \mathbf{W}_i^{ZCA}), i \in [0, k) \quad (5)$$

3.2.2 Shuffled Group Whitening

In order to further improve the quality of the sentence representation model, we proposed shuffled-group-whitening (SGW). We randomly divide the feature into multiple groups along the channel axis, and then perform ZCA whitening independently within each group. After whitening, we do a re-shuffled operation to recover features to their original arrangement. The process can be formulated as:

$$\mathbf{H} = \text{shuffled}^{-1}(\mathbf{G}\mathbf{W}(\text{shuffled}(\mathbf{Z}^T))) \quad (6)$$

This can bring two benefits. One is that it can avoid the limitation that only adjacent features can be put into the same group, so as to better decorrelation and then achieve better uniformity in the representation space. Another is that it brings a disturbance to samples, we can use it as a data augmentation method. Specifically, we repeat SGW multiple times and we can get different grouping results and then different whitened features. These "duplicated" features are different from each other, and thus increase the diversity of positive samples.

3.2.3 Connection to contrastive learning

We find that whitening and contrastive learning are not totally redundant but actually have some complementarity is non-trivial. Specifically, whitening decorrelates features through matrix decomposition, and makes the variance of all features equal to 1, that is, to project features into a spherical space. The "pushing" operation in contrastive learning is to approach a uniform spherical spatial distribution step by step through learning/iteration. Therefore, conceptually, whitening and contrastive learning are redundant in optimizing the uniformity of the

representation space. However, contrastive learning achieves uniformity by widening the distance between positive samples and all negative samples, but there is no explicit separation between negative samples. Whitening is the uniform dispersion of the entire samples, so there is complementarity between them. That is, whitening can supplement the lack of contrastive learning for the "pushing" operation between negative samples.

3.3 Multi-Positive Contrastive Loss

Since we get multi-positive samples from SGW module, however, the original contrastive loss in Eq. 1 is unable to handle multiple positives. We provide two possible options of contrastive loss which can adapt multi-positives. Given m positive samples, the objective function can be formulated as:

$$\mathcal{L}_1 = -\lambda_m \sum_{p=1}^m \log \frac{e^{\text{sim}(\mathbf{h}_i, \mathbf{h}_{i,p}^+)/\tau}}{\sum_{j=1}^N e^{\text{sim}(\mathbf{h}_i, \mathbf{h}_j^+)/\tau}} \quad (7)$$

$$\mathcal{L}_2 = -\log \sum_{p=1}^m \frac{\lambda_m e^{-\text{sim}(\mathbf{h}_i, \mathbf{h}_{i,p}^+)/\tau}}{\sum_{j=1}^N e^{\text{sim}(\mathbf{h}_i, \mathbf{h}_j^+)/\tau}} \quad (8)$$

where λ_m is a hyperparameter, it controls the impact of each positive. Eq. 7 puts the summation over positives outside of the log while Eq. 8 puts the sum of positives inside the log. It should be noted that in Eq. 8, there is a negative sign before the sum of positives, without it, the Eq. 8 will conduct hard mining, which means the maximum of $\sum_{p=1}^m e^{\text{sim}(\mathbf{h}_i, \mathbf{h}_{i,p}^+)/\tau}$ is mainly determined by $\max(e^{-\text{sim}(\mathbf{h}_i, \mathbf{h}_{i,p}^+)/\tau})$. If we add the negative sign, the loss function will be committed to punish the items with less similarity, which is good for bringing all positive samples closer to the anchor samples. In our framework, we adopt Eq. 7 as our final loss function because it can achieve better performance.

4 Experiments

4.1 Experiment Setup

In this section, We evaluate our method on seven Semantic Textual Similarity(STS) tasks and seven transfer tasks. We use the SentEval (Conneau and Kiela, 2018) toolkit for all of tasks.

Datasets. Semantic Textual Similarity(STS) tasks consist of seven tasks: STS 2012–2016 (Agirre et al., 2012, 2013, 2014, 2015, 2016),

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	56.70
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	63.73	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
ConSERT _{base}	64.64	78.49	69.07	79.72	75.95	73.97	67.31	72.74
SimCSE-BERT _{base}	68.40	82.41	74.38	80.91	78.56	76.85	72.23	76.25
DCLR-BERT _{base}	70.81	83.74	75.11	82.56	78.44	78.31	71.59	77.22
ArcCSE-BERT _{base}	72.08	84.27	76.25	82.32	79.54	79.92	72.39	78.11
* WhitenedCSE-BERT_{base}	74.03	84.90	76.40	83.40	80.23	81.14	71.33	78.78
ConSERT _{large}	70.69	82.96	74.13	82.78	76.66	77.53	70.37	76.45
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
ArcCSE-BERT _{large}	73.17	86.19	77.90	84.97	79.43	80.45	73.50	79.37
* WhitenedCSE-BERT_{large}	74.65	85.79	77.49	84.71	80.33	81.48	75.34	79.97
RoBERTa _{base} (first-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
* WhitenedCSE-RoBERTa_{base}	70.73	83.77	75.56	81.85	83.25	81.43	70.96	78.22

Table 1: Sentence embedding performance on STS tasks. We use the Spearman correlation to measure the relevance between gold annotations and scores predicted by sentence representations, we show the best performance in bold.

STS Benchmark (Cer et al., 2017) and SICK-Relatedness (Marelli et al., 2014). Each sample in those datasets has two sentences and a manually annotated similarity score from 0 to 5 to measure their similarity. The transfer tasks include MR (Pang and Lee, 2005), CR (Hu and Liu, 2004), SUBJ (Pang and Lee, 2004), MPQA (Wiebe et al., 2005), SST-2 (Socher et al., 2013), TREC (Voorhees and Tice, 2000), MRPC (Dolan and Brockett, 2005). In this tasks, we use a logistic regression classifier trained on top of the frozen sentence embeddings.

Baseline and competing methods. We compare WhitenedCSE against several classic methods on Semantic Textual Similarity datasets, *i.e.*, GloVe embeddings (Pennington et al., 2014), average BERT embeddings from the last layer (Devlin et al., 2018), BERT-flow (Li et al., 2020), BERT-whitening (Su et al., 2021), IS-BERT (Zhang et al., 2020), CT (Carlsson, 2021), ConSERT (Yan et al., 2021), SimCSE (Gao et al., 2021), as well as some most recent state-of-the-art methods, *i.e.*, MixCSE (Zhang et al., 2022a), ArcCSE (Zhang et al., 2021a), DCLR (Zhou et al., 2022).

Among these methods, SimCSE may be viewed as our direct baseline, because WhitenedCSE may be viewed as being transformed from SimCSE by adding the SGW and replacing the dual-positive contrastive loss with multi-positive contrastive loss.

Therefore, when conduct ablation study, we use SimCSE as our baseline.

Implementation details. We use the output of the MLP layer on top of the [CLS] as the our sentence representation. The MLP layer is consist of three components, which are a shuffled group whitening module, a 768×768 linear layer and a activation layer. Following SimCSE (Gao et al., 2021), we use 1×10^6 randomly sampled sentences from English Wikipedia as our training corpus. We start from pre-trained checkpoints of BERT (Devlin et al., 2018) and RoBERTa (Liu et al., 2019). At training time, we set the learning rate as $3e-5$, the batch size as 64. We train our model for 1 epoch with temperature $\tau = 0.05$. For BERT-base and BERT-large, we set the number of group size as 384, for RoBERTa-base and RoBERTa-large, we set the number of group size as 256. We set the number of positives as 3 for all of models. We evaluate the model every 125 training steps on the development set of STS-B, and keep the best checkpoint for evaluation on test sets. We conduct our experiments on two 3090 GPUs.

4.2 STS tasks

We conduct experiments on 7 semantic textual similarity(STS) tasks, and use SentEval toolkit (Conneau and Kiela, 2018) for evaluation. We use the

Model	MR	CR	SUBJ	MPQA	SST	TREC	MRPC	Avg.
GloVe (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
MoCoSE-BERT _{base}	81.07	86.43	94.76	89.70	86.35	84.06	75.86	85.46
ArcCSE-BERT _{base}	79.91	85.25	99.58	89.21	84.90	89.20	74.78	86.12
* WhitenedCSE-BERT _{base}	81.31	86.33	96.15	89.78	86.08	89.74	75.43	86.40
SimCSE-BERT _{large}	85.36	89.38	95.39	89.63	90.44	91.80	76.41	88.34
MoCoSE-BERT _{large}	83.71	89.07	95.58	90.26	87.96	84.92	76.81	86.90
ArcCSE-BERT _{large}	84.34	88.82	99.58	89.79	90.50	92.00	74.78	88.54
* WhitenedCSE-BERT _{large}	85.54	89.70	96.16	89.57	90.74	92.21	76.78	88.67

Table 2: Sentence embeddings performance on transfer tasks. We use the accuracy to measure the performance, and report the best in bold.

Spearman’s correlation coefficient as our evaluation metrics. The Spearman’s correlation uses a monotonic equation to evaluate the correlation of two statistical variables, it varies between -1 and 1 with 0 implying no correlation, and the closer the value is to 1, the closer the two statistical variables are to positive correlation.

Tab. 1 shows the evaluation results on 7 STS tasks, from which we can see that WhitenedCSE achieves competitive performance. Compared with SimCSE (Gao et al., 2021), WhitenedCSE achieves 2.53 and 1.56 points of improvement based on BERT_{base} and BERT_{large}. It also raise the performance from 76.57% to 78.22 % base on RoBERTa_{base}. Compared with recent works, WhitenedCSE also achieves the best performance in most of the STS tasks.

4.3 Transfer tasks

We also conduct experiments on 7 transfer tasks, and use SentEval toolkit (Conneau and Kiela, 2018) for evaluation. For each task, we train a logistic regression classifier on top of the frozen sentence embeddings and test the accuracy on the downstream task. In our experiment settings, we do not include models with auxiliary tasks, i.e., masked language modeling, for a fair comparison.

Tab. 2 shows the evaluation results. Compared with the SimCSE (Gao et al., 2021) baseline, WhitenedCSE achieves 0.59 and 0.33 accuracy improvement on average results based on BERT_{base} and BERT_{large}. Compared with recent works, WhitenedCSE also achieves the best performance in most of the transfer tasks, which further demonstrates the effectiveness of our method.

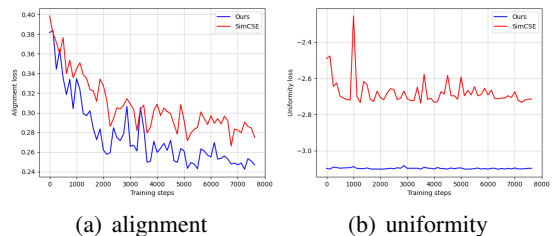


Figure 3: The alignment loss and the uniformity loss of WhitenedCSE and SimCSE. For both measures, lower number are better.

4.4 Alignment and Uniformity

In order to further quantify the improvement in uniformity and alignment of WhitenedCSE, we follow SimCSE (Gao et al., 2021), and use alignment loss and uniformity loss (Wang and Isola, 2020) to measure the quality of representations. Alignment is used to measure the expected distance between the embeddings of the positive pairs, and can be formulated as:

$$\ell_{\text{align}} = \mathbb{E}_{(x, x^+) \sim p_{\text{pos}}} \|f(x) - f(x^+)\|^2 \quad (9)$$

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

$$\ell_{\text{uniform}} = \log \mathbb{E}_{x, y \stackrel{i.i.d.}{\sim} p_{\text{data}}} e^{-2\|f(x) - f(y)\|^2} \quad (10)$$

We calculate the alignment loss and uniformity loss every 125 training steps on the STS-B development set. From Fig. 3, we can see that compared with SimCSE, WhitenedCSE performs better both on the alignment measure and the uniformity measure. We also find that the uniformity of our models

is well optimized at the beginning and remains stable throughout the training process. This further confirms that our method can improve the quality of sentence representation more effectively.

5 Ablation Analysis

In this section, we further investigate the effectiveness of our proposed model WhitenedCSE. For all experiments, we use BERT_{base} as our base model, and evaluate WhitenedCSE on the STS tasks unless otherwise specified.

5.1 Shuffling augments the positive samples

We prove theoretically and practically that SGW can be regarded as an effective data augmentation. We know different whitening transformations will get different whitened results, but all of them are representations for the same sample, so they can be regarded as positive samples for each other. In WhitenedCSE, we operate randomly shuffled on feature dimension, and divide the representations along the feature dimension into k groups. Since each time we use a different permutation, we can get a different representations \mathbf{Z} and the corresponding whitening matrix \mathbf{W} , We find it can be written as a form of feature-wise disturbance $\mathbf{Z}^* = \mathbf{Z} + \epsilon$:

$$\begin{aligned}\mathbf{Z}^* &= \mathbf{Z} + (\mathbf{W}^{ZCA} - \mathbf{1})\mathbf{Z} \\ &= \mathbf{Z} + (\mathbf{U}\mathbf{\Lambda}^{-1/2}\mathbf{U}^T - \mathbf{1})\mathbf{Z}\end{aligned}\quad (11)$$

Here, we treat $(\mathbf{U}\mathbf{\Lambda}^{-1/2}\mathbf{U}^T - \mathbf{1})\mathbf{Z}$ as a perturbation ϵ on the feature dimension. Thus, in WhitenedCSE, we use it as a data augmentation and generate more diverse positive samples. From Tab.4, we can see that, shuffling plays a very important role in the performance of the model.

5.2 The importance of Group Whitening

Recently, Su et al. (2021) directly apply whitening on the output of the BERT and have achieved remarkable performance at the time. This lead us to think whether whitening can be directly applied to the output of contrastive learning model to further improve the uniformity of the model representation space.

We consider two different whitening methods: PCA Whitening and ZCA Whitening. The difference between them is that ZCA Whitening uses an additional rotation matrix to rotate the PCA whitened data back to the original feature space,

Whitening strategy	STS-Avg.
None (unsup. SimCSE)	76.25
PCA-whitening	68.55
ZCA-whitening	72.11
Group-whitening	77.47
Shuffled-Group-whitening	78.78

Table 3: Performance comparison using different whitening methods on the test set of seven semantic textual similarity tasks. (Spearman’s correlation).

which can make the transformed data closer to the original input data.

$$\mathbf{W}^{ZCA} = \mathbf{U}_{rotate} \mathbf{W}^{PCA} \quad (12)$$

We use the in-batch sentence representations to calculate the mean value \bar{x} and the covariance matrix σ , and use the momentum to estimate the overall mean value μ and covariance matrix Σ .

$$\begin{aligned}\mu_n &= \beta\mu_{n-1} + (1 - \beta)\bar{x}_{n-1} \\ \Sigma_n &= \beta\Sigma_{n-1} + (1 - \beta)\sigma_{n-1}\end{aligned}\quad (13)$$

As the results shown in the Tab. 3, we found that directly applying the whitening transformation on contrastive learning models is detrimental to the performance. we attribute this to two reasons: (1) small batch size may not provide enough samples to obtain a suitable estimate for the full covariance matrix. (2) The covariance matrix obtained by high-dimensional features is not necessarily a positive definite matrix (maybe a semi-positive definite matrix), which may leads to errors in matrix decomposition. To alleviate this problem, we use the group whitening to control the extent of whitening. From Tab. 3 we can see that group whitening can significantly improve the performance.

5.3 Hyperparameters Analysis

For hyperparameters analysis, we want to explore the sensitivity of WhitenedCSE to these parameters. Concretely, we study the impact of the group size, the number of positive samples. We evaluate our model with varying values, and report the performances on the seven STS tasks.

The influence of group size. In WhitenedCSE, we divide the representation into k groups. However, we know the size of group controls the degree of whitening, and has a great effect on the effectiveness of WhitenedCSE, so we carry out an experiment with k varying from 32 to 384. As shown

Group Size	32	48	64	96
STS-Avg.	76.49	77.59	78.34	78.03
w/o shuffled	75.49	77.47	77.42	77.45
Group Size	128	192	256	384
STS-Avg.	<u>78.57</u>	78.15	77.97	78.78
w/o shuffled	77.47	77.45	77.42	77.43

Table 4: Effects of different group size and shuffling on seven STS tasks (Spearman’s correlation). The best performance and the second-best performance methods are denoted in bold and underlined fonts respectively.

Positive Number	2	3	4	5
STS-Avg.	77.81	78.78	78.65	78.74

Table 5: Effects of different positive number on seven STS tasks (Spearman’s correlation).

in Tab. 4, we can see that the best performance is achieved when $k = 384$, and the second best performance is achieved when $k = 128$. When k takes other values, the performance will drop slightly.

The influence of positive samples number. Sampling multi-positive samples can significantly enrich semantic diversity, we want to explore how the number of positive samples affect the performance of our model, so we conduct an experiment with positive number m varying from 2 to 5. From Tab. 5, we can see that when $m = 3$, our model achieve the best performance. However, due to the limitation of the memory size, we cannot exhaust all the possibilities, but we found that when $m \geq 2$, the performance of the model is always better than when $m = 2$, which confirms that multi-positive samples can bring richer semantics, allowing the model to learn better sentence representations.

The influence of different modules. In WhitenedCSE, using the proposed shuffled-group-whitening method is beneficial, and further using multi-hot positive samples brings additional benefit. We investigate their respective contributions to WhitenedCSE in Tab. 6. In our ablation study, we replace the whitening method with ordinary dropout technique, and still retain the multi-hot positive sample loss. Meanwhile, we use the proposed whitening method alone, and keep the number of positive samples 2. From Tab. 6, we find that multi-hot positive samples based on dropout only brings +0.12% improvement. This is reasonable because when the data augmentation is subtle (i.e., the dropout), using extra positive samples barely increases the diversity. In contrast, the proposed SGW gener-

Model	STS-Avg.
SimCSE-BERT-base	76.25
+ Shuffled Group Whitening	77.81 (+1.56)
+ Multiple Positives(k=3)	76.37 (+0.12)
+ Multiple Positives(k=4)	76.34 (+0.09)
WhitenedCSE-BERT _{base}	78.78 (+2.53)

Table 6: The performance on STS tasks when we use whitening module and multiple positives independently.

ates informative data augmentation, and thus well accommodates the multi-hot positive samples.

6 Conclusion

In this paper, we proposed WhitenedCSE, a whitening-based contrastive learning framework for unsupervised sentence representation learning. We proposed a novel shuffled group whitening, which reinforces the contrastive learning effect regarding both the uniformity and the alignment. Specifically, it retains the role of whitening in dispersing data, and can further improve uniformity on the basis of contrastive learning. Additionally, it shuffles and groups features on channel axis, and performs whitening independently within each group. This kind of operation can be regarded as a disturbance on feature dimension. We obtain multiple positive samples through this operation, and learn the invariance to this disturbance to obtain better alignment. Experimental results on seven semantic textual similarity tasks have shown that our approach achieve consistent improvement over the contrastive learning baseline.

Limitations

In this paper, we limit the proposed WhitenedCSE for sentence embedding learning. Conceptually, WhitenedCSE is potential to benefit contrastive learning on some other tasks, e.g., self-supervised image representation learning and self-supervised vision-language contrastive learning. However, we did not investigate the self-supervised image representation learning because this domain is currently dominated by masked image modeling. We will consider extending WhitenedCSE for vision-language contrastive learning when we have sufficient training resources for the extraordinary large-scale text-image pairs.

Ethics Statement

This paper is dedicated to deep sentence embedding learning and proposes a new unsupervised sentence representation model, it does not involve any ethical problems. In addition, we are willing to open source our code and data to promote the better development of this research direction

Acknowledgements

This work is supported by National Natural Science Foundation of China (No. 62293554) and the Fundamental Research Funds for the Central Universities (No. 226-2022-00051).

References

- Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Iñigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. 2015. [SemEval-2015 task 2: Semantic textual similarity, English, Spanish and pilot on inter-pretability](#). In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 252–263.
- Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. [SemEval-2014 task 10: Multilingual semantic textual similarity](#). In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 81–91.
- Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. [SemEval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation](#). In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 497–511. Association for Computational Linguistics.
- Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. [SemEval-2012 task 6: A pilot on semantic textual similarity](#). In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393.
- Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. [*SEM 2013 shared task: Semantic textual similarity](#). In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 32–43.
- Anthony J Bell and Terrence J Sejnowski. 1997. The “independent components” of natural scenes are edge filters. *Vision research*, 37(23):3327–3338.
- T. B. Brown, B. Mann, N. Ryder, M. Subbiah, and D. Amodei. 2020a. Language models are few-shot learners.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020b. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Fredrik Carlsson. 2021. Amaru cuba gyllensten, evangelia gogoulou, erik ylipää hellqvist, and magnus sahlgren. 2021. semantic re-tuning with contrastive tension. In *International Conference on Learning Representations*.
- Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. [SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation](#). In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14.
- 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.
- Alexis Conneau and Douwe Kiela. 2018. Senteval: An evaluation toolkit for universal sentence representations. *arXiv preprint arXiv:1803.05449*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- William B. Dolan and Chris Brockett. 2005. [Automatically constructing a corpus of sentential paraphrases](#). In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.
- Aleksandr Ermolov, Aliaksandr Siarohin, Enver Sangineto, and Nicu Sebe. 2021. Whitening for self-supervised representation learning. In *International Conference on Machine Learning*, pages 3015–3024. PMLR.
- Kawin Ethayarajh. 2019. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings. *arXiv preprint arXiv:1909.00512*.
- Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.
- K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

- Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning distributed representations of sentences from unlabelled data. *arXiv preprint arXiv:1602.03483*.
- Minqing Hu and Bing Liu. 2004. [Mining and summarizing customer reviews](#). In *ACM SIGKDD international conference on Knowledge discovery and data mining*.
- Tianyu Hua, Wenxiao Wang, Zihui Xue, Sucheng Ren, Yue Wang, and Hang Zhao. 2021. On feature decorrelation in self-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9598–9608.
- Lei Huang, Dawei Yang, Bo Lang, and Jia Deng. 2018. Decorrelated batch normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 791–800.
- Hervé Jégou and Ondřej Chum. 2012. Negative evidences and co-occurrences in image retrieval: The benefit of pca and whitening. In *European conference on computer vision*, pages 774–787. Springer.
- Taeuk Kim, Kang Min Yoo, and Sang-goo Lee. 2021. Self-guided contrastive learning for bert sentence representations. *arXiv preprint arXiv:2106.07345*.
- Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. *Advances in neural information processing systems*, 28.
- Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.
- Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196. PMLR.
- Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. *arXiv preprint arXiv:2011.05864*.
- 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. *arXiv preprint arXiv:1907.11692*.
- Lajanugen Logeswaran and Honglak Lee. 2018. An efficient framework for learning sentence representations. *arXiv preprint arXiv:1803.02893*.
- Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. [A SICK cure for the evaluation of compositional distributional semantic models](#). pages 216–223.
- Bo Pang and Lillian Lee. 2004. [A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts](#). pages 271–278.
- Bo Pang and Lillian Lee. 2005. [Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales](#). pages 115–124.
- Jeffrey Pennington, Richard Socher, and Christopher D 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.
- Ruizhi Qiao, Lingqiao Liu, Chunhua Shen, and Anton Van Den Hengel. 2016. Less is more: zero-shot learning from online textual documents with noise suppression. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2249–2257.
- Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.
- 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](#). pages 1631–1642.
- Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. 2021. Whitening sentence representations for better semantics and faster retrieval. *arXiv preprint arXiv:2103.15316*.
- Chih-Fong Tsai. 2012. Bag-of-words representation in image annotation: A review. *International Scholarly Research Notices*, 2012.
- Ellen M Voorhees and Dawn M Tice. 2000. [Building a question answering test collection](#). In *the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 200–207.
- Dong Wang, Ning Ding, Piji Li, and Hai-Tao Zheng. 2021. Cline: Contrastive learning with semantic negative examples for natural language understanding. *arXiv preprint arXiv:2107.00440*.
- Tongzhou Wang and Phillip Isola. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pages 9929–9939. PMLR.
- Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. [Annotating expressions of opinions and emotions in language](#). *Language resources and evaluation*, 39(2-3):165–210.
- Lei Wu, Steven CH Hoi, and Nenghai Yu. 2010. Semantics-preserving bag-of-words models and applications. *IEEE Transactions on Image Processing*, 19(7):1908–1920.

- Xing Wu, Chaochen Gao, Liangjun Zang, Jizhong Han, Zhongyuan Wang, and Songlin Hu. 2021. Esimcse: Enhanced sample building method for contrastive learning of unsupervised sentence embedding. *arXiv preprint arXiv:2109.04380*.
- Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021. Consert: A contrastive framework for self-supervised sentence representation transfer. *arXiv preprint arXiv:2105.11741*.
- Dejiao Zhang, Shang-Wen Li, Wei Xiao, Henghui Zhu, Ramesh Nallapati, Andrew O Arnold, and Bing Xiang. 2021a. Pairwise supervised contrastive learning of sentence representations. *arXiv preprint arXiv:2109.05424*.
- Dejiao Zhang, Wei Xiao, Henghui Zhu, Xiaofei Ma, and Andrew O Arnold. 2021b. Virtual augmentation supported contrastive learning of sentence representations. *arXiv preprint arXiv:2110.08552*.
- Shaofeng Zhang, Feng Zhu, Junchi Yan, Rui Zhao, and Xiaokang Yang. 2021c. Zero-cl: Instance and feature decorrelation for negative-free symmetric contrastive learning. In *International Conference on Learning Representations*.
- Yan Zhang, Ruidan He, Zuozhu Liu, Kwan Hui Lim, and Lidong Bing. 2020. An unsupervised sentence embedding method by mutual information maximization. *arXiv preprint arXiv:2009.12061*.
- Yanzhao Zhang, Richong Zhang, Samuel Mensah, Xudong Liu, and Yongyi Mao. 2022a. Unsupervised sentence representation via contrastive learning with mixing negatives.
- Yuhao Zhang, Hongji Zhu, Yongliang Wang, Nan Xu, Xiaobo Li, and Binqiang Zhao. 2022b. A contrastive framework for learning sentence representations from pairwise and triple-wise perspective in angular space. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4892–4903.
- Kun Zhou, Beichen Zhang, Wayne Xin Zhao, and Ji-Rong Wen. 2022. Debiased contrastive learning of unsupervised sentence representations. *arXiv preprint arXiv:2205.00656*.

ACL 2023 Responsible NLP Checklist

A For every submission:

- A1. Did you describe the limitations of your work?
Section Limitation
- A2. Did you discuss any potential risks of your work?
Since the number of pages in the text is limited and our model does not have significant potential risks, we do not discuss this.
- A3. Do the abstract and introduction summarize the paper’s main claims?
Section 1
- A4. Have you used AI writing assistants when working on this paper?
Left blank.

B Did you use or create scientific artifacts?

Section Reference

- B1. Did you cite the creators of artifacts you used?
Section Reference
- B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Because all artifacts we used are hereby granted, free of charge to any person.
- B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
The artifacts we used are not specified, and we don’t specify intended use for the artifacts we create.
- B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
The datasets we use are Semantic Textual Similarity datasets, They are not involved in the above content
- B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
This article mainly explains the main methods and experimental results of our model, its focus is not on the above content.
- B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
The data set we use is a very complete data set, and any relevant statistics are fixed and public

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a [question on AI writing assistance](#).

C Did you run computational experiments?

Section 4

- C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?

Section 4.1

- C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?

Section 4.1

- C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?

Section 4.2 and Section 5

- C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?

Section 4.1

D Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

- D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?

No response.

- D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?

No response.

- D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?

No response.

- D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?

No response.

- D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?

No response.