applied sciences

MDPI

Article

Grouped Contrastive Learning of Self-Supervised Sentence Representation

Qian Wang [1], Weiqi Zhang [1], Tianyi Lei [1] and Dezhong Peng [1,2,3,*]

[1] College of Computer Science, Sichuan University, Chengdu 610065, China; wangq@stu.scu.edu.cn (Q.W.); zhang_weiqi123@163.com (W.Z.); leity828@gmail.com (T.L.)
[2] Chengdu Ruibei Yingte Information Technology Co., Ltd., Chengdu 610054, China
[3] Sichuan Zhiqian Technology Co., Ltd., Chengdu 610065, China
* Correspondence: pengdz@scu.edu.cn

Abstract: This paper proposes a method called Grouped Contrastive Learning of self-supervised Sentence Representation (GCLSR), which can learn an effective and meaningful representation of sentences. Previous works maximize the similarity between two vectors to be the objective of contrastive learning, suffering from the high-dimensionality of the vectors. In addition, most previous works have adopted discrete data augmentation to obtain positive samples and have directly employed a contrastive framework from computer vision to perform contrastive training, which could hamper contrastive training because text data are discrete and sparse compared with image data. To solve these issues, we design a novel framework of contrastive learning, i.e., GCLSR, which divides the high-dimensional feature vector into several groups and respectively computes the groups' contrastive losses to make use of more local information, eventually obtaining a more fine-grained sentence representation. In addition, in GCLSR, we design a new self-attention mechanism and both a continuous and a partial-word vector augmentation (PWVA). For the discrete and sparse text data, the use of self-attention could help the model focus on the informative words by measuring the importance of every word in a sentence. By using the PWVA, GCLSR can obtain high-quality positive samples used for contrastive learning. Experimental results demonstrate that our proposed GCLSR achieves an encouraging result on the challenging datasets of the semantic textual similarity (STS) task and transfer task.

Keywords: contrastive learning; self-attention; data augmentation; grouped representation; unsupervised learning

check for updates

Citation: Wang, Q.; Zhang, W.; Lei, T.; Peng, D. Grouped Contrastive Learning of Self-Supervised Sentence Representation. *Appl. Sci.* **2023**, *13*, 9873. https://doi.org/10.3390/app13179873

Academic Editor: Chilukuri K. Mohan

Received: 21 July 2023
Revised: 28 August 2023
Accepted: 30 August 2023
Published: 31 August 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Representation learning of sentences involves learning a meaningful representation for a sentence. Most downstream tasks in natural language processing (NLP) are implemented with sentence representation [1–5].

Recently, researchers have achieved great advances in sentence representation based on contrastive learning with pre-trained language models [6–10]. On the one hand, the large-scale pre-trained language models (PLMs), typified by BERT [11], are trained with unlabeled data, improving the state-of-the-art results in most downstream tasks. Therefore, PLMs are applied to various real scenarios, such as text generation [8], name entity recognition [12], question answering [13], and translation [13]. On the other hand, unsupervised representation learning based on contrastive learning advances the development of computer vision [14–17]. Therefore, many researchers combine PLMs with contrastive learning to conduct sentence representation tasks [18,19]. For example, Wu et al. [20] adopt back-translate as the data augmentation method to produce positive samples used for contrastive learning and PLMs as the backbone to obtain semantic feature of sentences, achieving a promising result for sentence representation. Gao et al. [21] respectively take the standard dropout

mask of the transformer and cosine similarity as the data augmentation method and as the contrastive objective function to conduct the contrastive training, producing a meaningful sentence representation.

However, there are issues with implementing contrastive learning in sentence representation: (a) An appropriate data augmentation method is needed to produce positive samples used for contrastive learning. In contrastive training, the semantic similarity between the positive example pair should be narrow. Therefore, improper data augmentation may change the semantic information of sentences, resulting in difficulties advancing performance. (b) The information of text data are sparse and discrete. Unlike image data, the information between the adjacent pixels is continuous, while the information of text data is discrete, indicating that the model could not learn the distinguishing features by contrastive learning. (c) Similarity computing between high-dimensional vectors could lose the local information of vectors. Generally, the objective of contrastive learning is to minimize the similarity of high-dimensional vectors, which could not make use of local information of vectors well and could affect performance.

To solve the issues above, we propose Grouped Contrastive Learning of self-supervised Sentence Representation (GCLSR). GCLSR adopts continuous data and partial data augmentation to obtain high-quality positive samples used for contrastive learning. Due to the discrete and sparse text data, GCLSR designs a self-attention mechanism to focus on informative words by measuring the importance of every word in a sentence. To address high-dimensional feature vectors, GCLSR proposes grouped contrastive learning to disentangle more local information of feature vectors.

The contributions of this paper are summarized as follows:

- We propose a new data augmentation method called partial-word vector augmentation (PWVA) to obtain positive samples used for contrastive learning. PWVA performs data augmentation on partial word vectors of the word embedding space of a sentence. In this way, the positive sample pairs can retain more original semantic information, which could enhance and facilitate contrastive learning.
- We design a new computation method of self-attention to help the model focus on the informative words of a sentence. Experimental results show that the use of self-attention can enhance the representation of discrete and sparse text data.
- We design a new paradigm of contrastive learning called the Grouped Contrastive Learning of self-supervised Sentence Representation (GCLSR), which can make use of more local information of high-dimensional feature vectors.
- We evaluate GCLSR on different datasets. Experimental results demonstrate that our proposed GCLSR achieves a promising result on sentence representation. Additionally, we further investigate effectiveness of the GCLSR through an ablation study and explore possible implementation schemes based on our method.

The rest of this paper is organized as follows: The related works on representation learning based on contrastive learning, text data augmentation, and self-attention are introduced in Section 2. Our proposed GCLSR is presented in Section 3. Sections 4 and 5 respectively evaluate and investigate GCLSR. Conclusions and future work are presented in Section 6.

2. Related Work

2.1. Representation Learning Based on Contrastive Learning

Contrastive learning obtains promising results in representation learning [14,17,22]. Generally, a Siamese network is used to construct the contrastive framework and conduct contrastive training [14].

In the computer vision domain, contrastive learning achieves significant improvement in the representation of image. SimCLR [14] uses an encoder and projection head as the contrastive framework, which advances the state-of-the-art results for image representation. BYOL [17] designs a momentum encoder to avoid collapsing on contrastive training, which obtains an encouraging result. SimSiam [16] uses BYOL's contrastive framework, but it

changes the way that a network's parameters are updated. Surprisingly, stop-gradient not only solves the issue of the model's collapse but also obtains comparable results in STS tasks.

Contrastive learning achieves promising results for image representation. Therefore, researchers have started to adopt contrastive learning to obtain high-quality sentence representation. CERT [23] augments a sentence by back-translation and performs contrastive training using the contrastive framework of MoCo. CMV-BERT [24] adopts different tokenizers to conduct sentence augmentation and performs contrastive training on the framework of the SimSiam. ConSERT [25] performs data augmentation (such as token shuffling, adversarial attack, cutoff, and dropout) on word vectors of BERT to obtain positive samples and achieve encouraging results. More details about contrastive learning in sentence representation are shown in Table 1.

Table 1. Some contrastive learning methods in sentence representation. P: the computing of loss just includes positive samples. P + N: the computing of loss includes positive and negative samples, i.e., $\ell = -log\frac{exp(sim(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} exp(sim(z_i, z_k)/\tau)}(k \neq i)$, where z_i, z_j, and z_k are the positive samples, while z_k represents the negative samples. τ is a temperature parameter. $sim(\cdot)$ denotes the similarity. Discrete/continuous and full: data augmentation is discrete/continuous and performed on every word of a sentence.

Model	Backbone	Data Augmentation	Loss	Framework
CERT [23]	Pre-trained BERT	Back-translation (Discrete and Full)	P + N	Based on MoCo
CMV-BERT [24]	ALBERT (3 layers)	Multi-tokenizers (Discrete and Full)	P	Based on SimSiam
CLEAR [20]	Transformer	Substitution (Discrete and Full)	P + N	Based on SimCLR
ConSERT [25]	Pre-trained BERT	Dropout (Continuous and Full)	P + N	Based on SimCLR
SimCSE [21]	Pre-trained BERT	Dropout (Continuous and Full)	P + N	Based on SimCLR

While great success has been achieved by contrastive learning in sentence representation, deficiencies still exist in the aforementioned methods, hampering the improvement of performance. The details are shown below: (1) Most methods directly utilize the framework of computer vision as the pipeline for contrastive learning. Therefore, it could hamper contrastive training because text data are discrete and sparse compared to image data. (2) The well-performing pre-trained models (such as BERT) are adopted as the backbone network of contrastive learning. Consequently, they cannot evaluate the performance of a lightweight model on sentence representation using contrastive learning. After all, the pre-trained model works well in NLP tasks. (3) Improper data augmentation could change the original semantics of a sentence. Most methods use discrete data augmentation to produce positive samples to perform contrastive training, which could deteriorate the original semantics. Different from the aforementioned methods, we design a dedicated contrastive learning framework for sentence representation, namely, GCLSR. To obtain high-quality positive samples, GCLSR uses partial-word vector augmentation, a continuous form of data augmentation, which can maintain more of the original semantics of sentences. Further, GCLSR uses a lightweight model TextCNN to explore the effectiveness of contrastive learning on sentence representation.

2.2. Text Data Augmentation

Data augmentation is an effective strategy to improve performance and steadiness of training. Wei et al. [26] proposed a popular data augmentation method called EDA for text classification and achieved promising results. Wang et al. [27] use k-nearest neighbor word vectors as the positive samples. Guo et al. [28] obtain positive sample pairs by performing a linear interpolation between word vectors.

While many data augmentation methods obtain encouraging results in NLP, there is no dedicated one for contrastive training. Generally, the positive samples used for contrastive training are produced by data augmentation. Therefore, the above-mentioned approaches

could not be directly employed to generate positive samples. The explanations for this are listed as follows: (1) A high semantic similarity should be reserved between positive samples. On the contrary, the contrastive model could be collapsing easily. (2) A data augmentation method could be implemented on partial data to produce positive samples. In this way, more original semantics could be preserved between positive samples, which could help the model learn to distinguish features easily. (3) The augmentation of text is as continuous as possible. Most methods above, such as EDA and back-translation, are discrete, which may make the contrastive training unsteady and hurt the generalization of model. Different from the existing data augmentation methods, our proposed PWVA is a continuous data augmentation strategy. PWVA conducts data augmentation for partial words of a sentence in the word embedding space, which can preserve more original semantics between positive samples and facilitate the contrastive training.

2.3. Self-Attention in Language Model

Great progress has been made in the development of attention since Bahdanau et al. [29] adopted attention to enhance the performance of NLP tasks. Devlin et al. [11] designed an encoder with attention in order to process sentences and achieved great performance on the various tasks of NLP. However, it needs additional operations (such as position-wise feed-forward networks and layer normalization) to ensure steady training, resulting in difficulties in application to a practical, lightweight computational platform. Different from the method proposed by Devlin [11], we design a self-attention mechanism with low computing consumption to compute the importance of words in a sentence without any additional operations. In addition, to help the lightweight model measure the importance of a word for a sentence, we rewrite the computing process of sef-attention slightly. In this way, the use of self-attention in contrastive learning can help the model focus on the informative words of a sentence. The details of our proposed method for self-attention are shown in Section 3.

3. Methodology

As discussed above, contrastive learning can be conducted by mainly obtaining positive samples and designing a contrastive framework. In this paper, we propose a Grouped Contrastive Learning of self-supervised Sentence Representation (GCLSR). Figure 1 illustrates the overall architecture and training pipeline of GCLSR. As shown in Figure 1, GCLSR contains three parts: partial-word vector augmentation (introduced in Section 3.1), self-attention (introduced in Section 3.2), and the GCLSR network (introduced in Section 3.3). The upper right plot includes the details of the GCLSR network. The lower right plot is the visualization of PWVA (introduced in Section 3.1).

3.1. Partial Word Vector Augmentation

As discussed above, performing data augmentation in contrastive learning is done in order to obtain positive samples. However, most existing methods are discrete and performed on full words of a sentence, which could deteriorate original semantic information for discrete and sparse text data. Therefore, we design a continuous and partial-word vector augmentation (PWVA) for contrastive learning. Furthermore, a word vector is a vector with fixed dimensionality, and every element in a word vector is a real value. Therefore, a word vector can be treated as a 1D discrete signal. In this way, word vectors can be processed by strategies of digital signal processing. Our proposed PWVA is based on this insight in order to implement data augmentation. To be exact, PWVA is conducted by two probability choices. Let $W = \{w_i \in \mathbb{R}^d\}_{i=1}^N$ be the N word vectors with d dimensionality. The first probability choice of PWVA is represented by:

$$w_{aug} = \rho(A_{gwn}(w_i), A_{rzs}(w_i), A_{ifft}(w_i), A_{rbn}(w_i); p1, p2, p3, p4), \quad (1)$$

where $\rho(\cdot)$ is a function aiming to choose data augmentation strategies from Random Zero Setting (RZS), Inverse Fast Fourier Transformation (IFFT), Gaussian White Noise (GWN), and Random Background Noise (RBN) by the probabilities $p1, p2, p3$, and $p4$, respectively. The w_{aug} denotes the augmented word vectors. The second choice of PWVA can be expressed below:

$$w_{pwva} = \varrho(w_{aug}, w_i; p), \tag{2}$$

where ϱ is a function to select the final PWVA output w_{pwva} from w_{aug} and w_i, with probability p. The visualization of PWVA is shown in the lower right plot of Figure 1. In addition, four data augmentation strategies employed in Equation (1) are explained below.

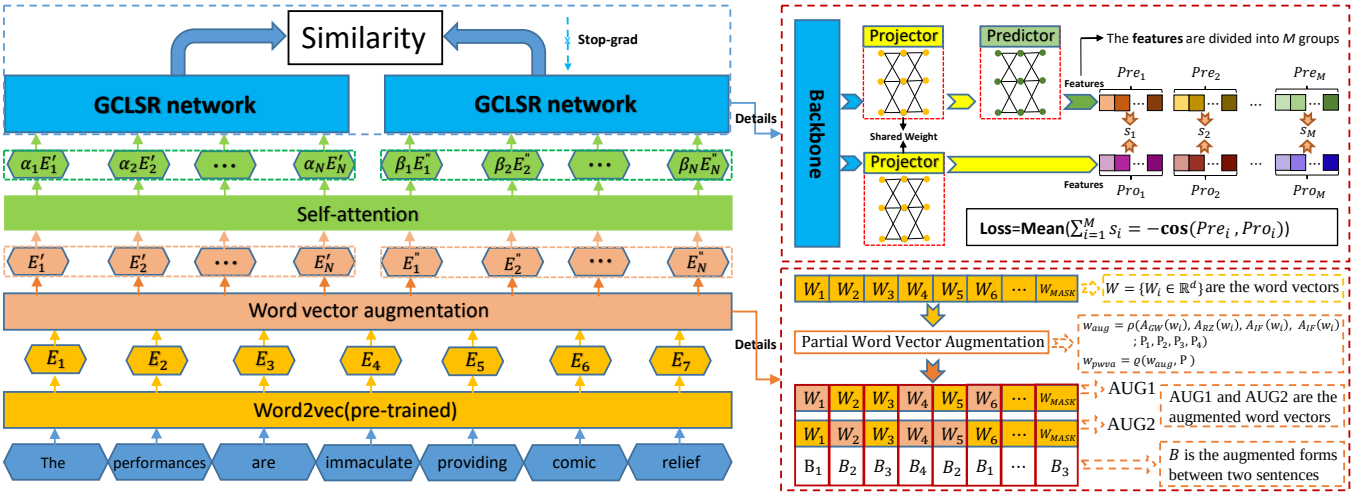

Figure 1. The GCLSR architecture.

- **Gaussian White Noise (GWN)**
  In order to improve the robustness of our model, we introduce Gaussian white noise (as illustrated in Figure 2a) into the word vectors. This approach is inspired by the work of Uchaikin et al. [30]. Gaussian white noise can be mathematically represented as follows:

$$A_{gwn}(w_i) = w_i + \lambda \cdot \mathcal{N}(0, 1), \tag{3}$$

  where λ represents the trade-off parameter, while $\mathcal{N}(0, 1)$ refers to the standard normal distribution.
- **Random Zero Setting (RZS)**
  To mitigate data dependence and enhance generalization ability, we employ a technique called random zero setting $A_{rzs}(w_i) = Dropout(w_i)$ (as illustrated in Figure 2b). This technique enables us to randomly assign zero values to certain word vector components.
- **Inverse Fast Fourier Transformation (IFFT)**
  To extract features in the frequency domain, we utilize word vectors and subsequently apply the inverse fast Fourier transform (IFFT) as illustrated in Figure 2c to convert them into the time domain. The word vectors undergo slight modifications after undergoing the IFFT process, thereby enhancing the resilience of the data boundary. The mathematical representation of the IFFT can be expressed as follows:

$$A_{ifft}(w_i) = Real(IFFT(FFT(w_i))), \tag{4}$$

  where $Real(\cdot)$ denotes the real part.
- **Random Background Noise (RBN)**
  Random background noise cannot be learned by a model, as stated in the research conducted by [31]. Therefore, to enhance training stability, we introduce random

background noise into the word vectors, as depicted in Figure 2d. The formulation for random background noise (RBN) is given below:

$$A_{rbn}(w_i) = w_i + uniform(0, 0.1), \tag{5}$$

where $uniform(\cdot)$ is the uniform distribution.

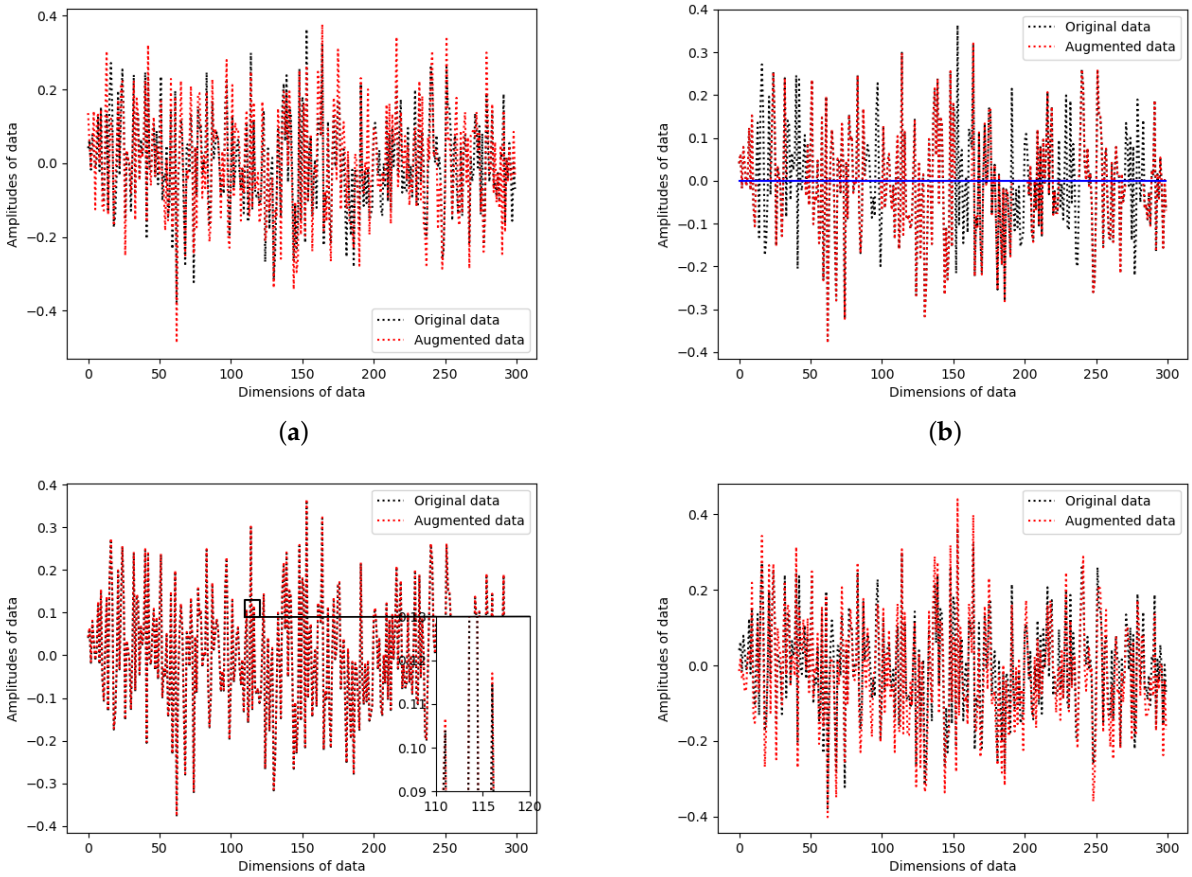

Figure 2. Four word vector augmentation methods of the PWVA. Note that we perform data augmentation on word vectors of a sentence rather than on the original sentence. (**a**) Gaussian White Noise: Amplitudes of noise added to the word vectors follow the standard normal distribution. (**b**) Random Zero Setting: The data corresponding to the black curve are set to zero. (**c**) Inverse Fast Fourier Transformation:Differences from the data boundary can be observed in the right zoom window. (**d**) Random Background Noise: Amplitudes of noise added to the word vectors follow a uniform distribution.

In summary, continuous and PWVA can enhance and facilitate contrastive learning. The characteristic of "continuous" PWVA can ensure that there is no semantic gap in word vectors, while the "partial" can retain more original semantics of word vectors. In this way, a model can readily acquire a richer set of discriminative features by assimilating the disparities between the initial word vectors and their respective augmented counterparts. In contrast, all existing methods obtain distinguishing features between augmented data, resulting in difficulties in contrastive training. This insight is the main contribution of PWVA. In addition, as shown in the lower right plot of Figure 1, we present a visualization of the PWVA process. Specifically, we apply data augmentation twice to the word vector space using PWVA to obtain two sets of positive sample pairs, AUG1 and AUG2. In AUG1 and AUG2, the orange boxes represent the augmented word vectors, while the yellow

boxes indicate that the word vectors have not been augmented. As a result, there are four possible combinations, B1, B2, B3, and B4, between AUG1 and AUG2. B1 represents the scenario where the word vector W1 in AUG1 is augmented, while W1 in AUG2 remains unchanged. The term "partial" indicates that some word vectors in both AUG1 and AUG2 are not augmented, thus preserving more of their original semantics for contrastive learning. The results of the ablation study are presented in Section 5.

3.2. Self-Attention of the Word Vectors

We perform PWVA to obtain high-quality positive samples. However, for a lightweight model, it could not effectively capture the importance of a word in a sentence. Therefore, we design a self-attention mechanism to capture the importance of words and facilitate contrastive training. Self-attention is applied to many scenarios and achieves great success. Inspired by the work of [11], we design a dedicated self-attention method to help the model focus on informative word vectors from the discrete and sparse text data. Note that the word vectors are produced by the pre-trained word2vec [32] before carrying out data augmentation. Hence, the word vectors already include some semantic information. Furthermore, the self-attention added to the word vectors can make the model focus on the features useful for distinguishing semantic information. The details are shown in Figure 3. Note that our method of self-attention is different from BERT's. The main differences are as follows: (1) We fill the value 1×10^{-9} after computing the scores for padding tokens. (2) We first compute the importance of the words to a sentence, and then multiply it by the original word vector. More details are shown in Algorithm 1. To verify the effectiveness of our method, we conduct an experiment to compare the performance on an STS task with the state-of-the-art models proposed by [11]. We observe that our proposed method increases the average Spearman's correlation from 62.97 to 66.75 (+3.78) with the same time complexity $O(n^2)$. In addition, we visualize the process of our proposed self-attention method. As shown in Figure 3, let $N =$ be the number of words in a sentence. $MASK$ and $s_{i,j} = x_i * x_j$ are the mask matrix (the value of which equals 0 if the word is the padding token) and attention of the word x_i to x_j in a sentence, respectively. Specially, $S = \sum_{j=1}^{N} s_{i,j}$ $(i = 1, 2, \cdots, N)$ can represent the importance of the word x_i to a sentence. We can observe from Figure 3 that the first few words with a larger value of "importance" are the word "comic", "relief", "performances", and "immaculate", which can help the model watch for the crucial information in a sentence.

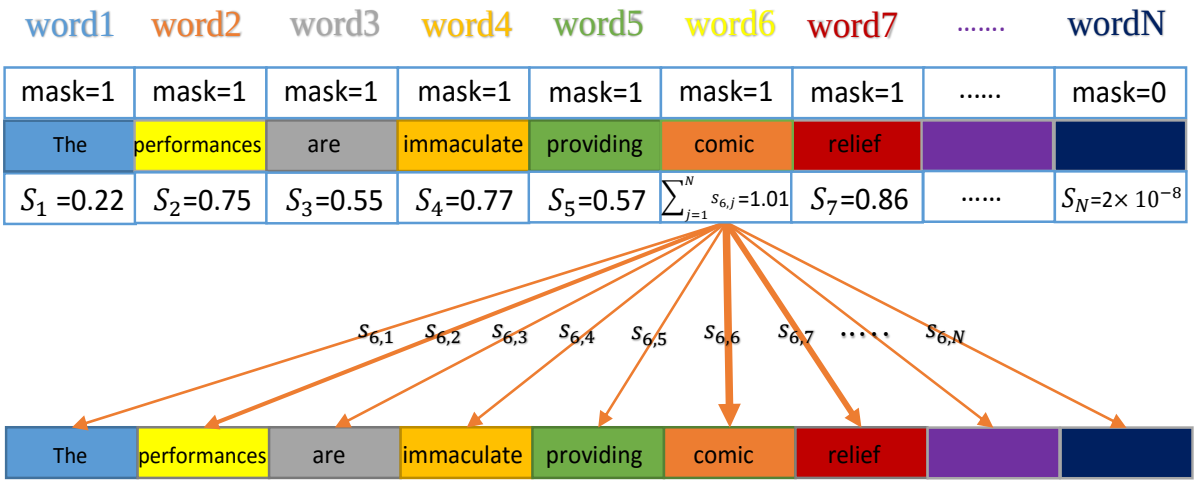

Figure 3. The self-attention of word vectors.

Algorithm 1: Self-attention of Word Vectors

Input: S: a sentence with N word vectors and dimensionality d_k
Output: attention of word vector
initialization: Q ← K ← V ← W
#compute the mask matrix
for *w in S* **do**
  if *w is padding* **then**
    mask ← 0;
  else
    mask ← 1;
  end
end
#compute the words' importance
sim←$\frac{Q \cdot K^T}{\sqrt{d_k}}$;
for *w, m in (sim, mask)* **do**
  if *m==0* **then**
    w ← 1×10^{-9};
  end
  s ← w;
end
scores←$\sum_{j=1}^{N} s_{i,j}$;
return scores· V

3.3. Grouped Contrastive Learning

By conducting PWVA and self-attention, the construction of positive samples is finished. Next, we introduce grouped contrastive learning to obtain sentence representation. Generally, the pipelines of contrastive learning are that one first performs data augmentation to produce positive samples, then obtains the features computed by the backbone, and finally computes the contrastive loss [16]. Unfortunately, computing a contrastive loss between high-dimensional vectors could not make use of the local information of vectors well. To solve this issue, we propose the GCLSR to mitigate the aforementioned drawback during contrastive training. As shown in Figure 1, the GCLSR consists of two branches. The first branch includes the backbone, projector [14], and predictor [17], while the other is the backbone and projector. In particular, in order to make use of local information about features, we first divide the features of the projector and predictor into M groups with D dimensionality. The grouped features of the projector and predictor can be denoted as $Fea_{Pro} = \{Pro_i \in \mathbb{R}^D\}_{i=1}^{M}$ and $Fea_{Pre} = \{Pre_i \in \mathbb{R}^D\}_{i=1}^{M}$, respectively. Finally, we use the negative mean of the cosine similarity as the contrastive loss [17]:

$$\ell = -Mean(\sum_{i=1}^{M}(\frac{Pre_i \cdot Pro_i}{\|Pre_i \cdot Pro_i\|_2})), \tag{6}$$

where $\|\cdot\|$ is l_2-norm. In addition, we adopt the symmetrical loss to improve performance:

$$\ell_{sym} = -\frac{1}{2}Mean(\sum_{i=1}^{M}(\frac{Pre_i \cdot Pro_i}{\|Pre_i \cdot Pro_i\|_2} + \frac{Pro_i \cdot Pre_i}{\|Pre_i \cdot Pro_i\|_2}). \tag{7}$$

4. Experiments

We systematically assess the efficacy of our novel approach, denoted as GCLSR, across seven distinct tasks focused on semantic textual similarity (STS). Moreover, we rigorously examine its performance on an additional set of seven transfer tasks. Worth highlighting is our deliberate choice of a lightweight model—TextCNN—as the foundational architecture.

This decision allows us to meticulously probe the potential of contrastive learning in enhancing sentence representations. It is pertinent to underscore that our intention is not to draw comparisons to the prevailing state-of-the-art benchmarks. Furthermore, it is essential to emphasize that both the STS experiments and the transfer tasks are conducted under a fully unsupervised setting. Notably, during the training phase, no STS datasets—comprising training, validation, or test sets—are employed. This approach ensures the integrity of our experimental setup and validates the intrinsic strength of our proposed methodology.

4.1. Implementation Settings

Unless explicitly stated otherwise, we adhere to the ensuing configuration for the pre-training phase of our contrastive self-supervised methodology:

- **Backbone**. We use TextCNN [33] as the default backbone. Specifically, the filter region size is [1,1,1,6,15,20]. The number of filters is 300. Note that we do not use a fully-connected (FC) layer or dropout at the end of the backbone, because this makes the results worse.
- **Projector**. The projection layers include three FC layers. Every output of an FC layer has a batch normalization [34] and ReLU, except for the last FC layer. The dimension of the hidden and output layers is 4096.
- **Predictor**. The prediction layers have two FC layers. The hidden layers have a batch normalization (BN) and ReLU, while the output layers do not have BN and ReLU. The hidden and output layers are both endowed with dimensions of 1024 and 4096, respectively, resulting in a bottleneck architecture that substantially enhances the model's robustness [16].
- **Optimizer**. The SGD is used for the optimizer. The learning rate (LR) is $base\_lr * BatchSize/128$ (the $base\_lr$ is 0.03). The LR has a cosine decay schedule [35]. The weight decay is 0.001. We also use the warm-up (5 epochs). Additionally, the momentum is 0.9 before warm-up epochs and 0.8 after warm-up epochs, which makes the model more robust (more details are shown in Section 5).

4.2. Semantic Textual Similarity Task

The goal of the semantic textual similarity task (STS) is to evaluate the similarity between two sentences by directly computing the cosine distance [36]. Then, the cosine distance correlates with a labeled similarity score (from 0 to 5) by Pearson or Spearman correlations to obtain a matching score. In this way, the matching score can reflect the semantic similarity between two sentences. We train our self-supervised GCLSR model with pre-trained word2vec on 10^4 sampled sentences randomly drawn from English Wikipedia [21]. The stop epochs are 20, and the best checkpoint on validation datasets is used for testing. Finally, we use the SentEval toolkit [36] to measure our proposed method on 7 STS tasks, i.e., STS 2012–2016 [37–41], STS Benchmark [42], and SICK-Relatedness [43]. In these datasets, sentence pairs are from news articles, news conversations, forum discussions, headlines, and image and video descriptions. Following [16], (a) we employ Spearman correlation as the only metric to evaluate the quality of sentence representation in STS tasks. Ref. [16] argues that Spearman correlation better suits the needs of evaluation; (b) no additional networks are applied on top of sentence representation. Put differently, we directly calculate the Spearman correlation using cosine similarities; (c) given that STS data of every year include several sub-datasets, we concatenate all sub-datasets to calculate the Spearman correlation. The operation "concatenate", incorporating different subsets, is more proper compared with other methods in practical applications.

The results of the evaluation are shown in Table 2. We observe that all variations we proposed work well and are better than word2vec embeddings (on average). Specifically, we improve the average Spearman correlation from 44.16 to 66.75 compared with word2vec embeddings (on average), the pre-trained language model BERT (from 56.57 to 66.75), RoBERT (from 56.57 to 66.75), and CLEAR (from 61.8 to 66.75). Furthermore, our proposed data augmentation PWVA outperforms the compared methods of EDA and back-translation on STS

tasks using the proposed contrastive learning framework GCLSR$_{base}$. The application of self-attention (GCLSR$_{base}$+PWVA+self-att.) and grouping (GCLSR$_{base}$+PWVA+self-att.+GP) can also improve the performance. The experimental results show that our proposed method GCLSR$_{base}$+PWVA+self-att.+GP can also achieve a better result compared with GCLSR$_{base}$+EDA+self-att.+GP and with GCLSR$_{base}$+TransL.+self-att.+GP. In addition, we note a significant distinction wherein strong performance in the STS tasks does not inherently translate into improved results in the transfer tasks. Consequently, it is prudent to primarily consider the outcomes from the STS evaluations for the purpose of comparison.

Table 2. The evaluation of sentence representation in STS tasks. All results are computed with the Spearman correlation. *: results from [21]; **: results from [20]; the remaining results are evaluated by us. GCLSR$_{base}$ means that the model only consists of a backbone, projector, and predictor. The model receives the same two word vectors as the input, i.e., no data augmentation is used. EDA is a text data augmentation method proposed by [26]. TransL denotes the data augmentation back-translation. Self-att and GP denote the self-attention mechanism and feature grouping, respectively.

Model	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
Word2vec embeddings(avg.)	33.75	43.20	36.95	55.23	54.85	36.24	48.90	44.16
BERT$_{base}$(first-last avg.) *	39.70	59.38	49.67	66.03	66.19	53.87	62.06	56.70
RoBERT$_{base}$(first-last avg.) *	40.88	58.74	49.07	65.63	61.48	58.55	61.63	56.57
CLEAR **	49.00	48.90	57.40	63.60	65.60	**75.60**	**72.50**	61.80
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	63.73	66.28
GCLSR$_{base}$	57.47	68.70	64.03	72.84	67.90	65.64	59.55	65.16
GCLSR$_{base}$+EDA	57.33	68.09	63.65	72.01	66.63	65.34	59.71	64.68
GCLSR$_{base}$+TransL	**60.53**	66.09	63.50	72.83	67.39	66.09	59.78	65.17
GCLSR$_{base}$+PWVA	58.92	67.94	64.41	73.54	68.72	66.16	**59.85**	65.65
GCLSR$_{base}$+PWVA+self-att.	57.62	71.00	65.83	75.51	69.81	67.41	59.41	66.66
GCLSR$_{base}$+EDA+self-att.+GP	58.13	68.85	64.14	73.40	66.92	65.31	59.61	65.19
GCLSR$_{base}$+TransL.+self-att.+GP	58.80	68.00	64.70	73.74	68.50	67.24	59.67	65.81
GCLSR$_{base}$+PWVA+self-att.+GP	57.81	**71.01**	**65.83**	**75.62**	**70.01**	**67.58**	59.34	**66.75**

4.3. Transfer Task

Transfer task is used to evaluate the performance of downstream tasks using sentence representation [36]. Generally, a classifier is added on the top layer of a sentence representation model to evaluate the performance of the transfer task. Note that the classifier (consisting of linear layers) can be trained, while the sentence representation model needs to be frozen. Our proposed method underwent rigorous testing across a spectrum of tasks, including MR [44], CR [45], SUBJ [46], MPQA [47], SST-2 [48], TREC [49], and MRPC [50]. The pre-trained stage is the same as for STS tasks. The evaluation results are shown in Table 3. We find that the overall tendency of results is the same as STS tasks. However, there are two abnormalities that need to be explained. (1) The pre-trained model BERT$_{base}$ obtains a better result compared with our proposed model on transfer tasks. Firstly, we only chose 10^4 sentences from the wiki to perform the pre-training. Secondly, the number of parameters of our model is far less than BERT$_{base}$. Therefore, we take a short time to perform pre-training (about 1 h on 1 Tesla v100 GPU). (2) The application of self-attention and grouping harms the performance slightly compared with GCLSR$_{base}$+PWVA. A possible explanation is that the implementation of PWVA is on word vectors, which could change the original semantic information of vectors. In addition, we do not perform joint training with tasks, which means the model could not digest the learned contrastive features.

Table 3. The results of transfer tasks. All results are computed with the Spearman correlation. *: results from [21]; the remaining results are evaluated by us.

Model	MR	CR	SUBJ	MPQA	SST-2	TREC	MRPC	Avg.
Word2vec embeddings(avg.)	75.91	77.56	89.31	87.18	80.89	77.40	**72.17**	80.06
$BERT_{base}$(first-last avg.) *	**78.66**	**86.25**	**94.37**	**88.66**	**84.40**	**92.80**	69.54	**84.94**
$GCLSR_{base}$	76.78	79.02	90.21	88.35	81.77	83.00	73.28	81.77
$GCLSR_{base}$+EDA	76.56	79.55	90.54	88.54	81.66	84.40	72.99	82.03
$GCLSR_{base}$+TransL	76.61	79.52	90.41	88.74	81.27	84.00	73.04	81.94
$GCLSR_{base}$+PWVA	76.76	**80.08**	90.66	88.59	81.60	85.20	73.57	**83.35**
$GCLSR_{base}$+PWVA+self-att.	76.97	78.81	**90.98**	88.36	80.51	84.60	**73.74**	82.00
$GCLSR_{base}$+EDA+self-att.+GP	76.90	79.58	90.57	88.50	**81.82**	**85.60**	72.75	82.25
$GCLSR_{base}$+TransL.+self-att.+GP	76.98	79.32	90.41	88.54	81.16	84.00	73.28	81.96
$GCLSR_{base}$+PWVA+self-att.+GP	**77.69**	79.87	90.89	**88.99**	81.44	83.80	73.39	82.30

5. Further Investigation of GCLSR

We design a novel contrastive learning paradigm, namely, GCLSR, that consists of three crucial components, i.e., (a) data augmentation, (b) self-attention, and (c) grouped contrastive learning, to study the performance of contrastive learning on sentence representation. Experimental results show that our proposed GCLSR achieves a promising result. However, some experimental settings of GCLSR influence the performance of sentence representation, such as warm-up, weight decay, etc. Therefore, we conduct ablation experiments to analyze them further. All experiments are conducted in STS 2012–2015.

5.1. Effect of Batch Size

Given that a large batch size could impact the performance shown in previous works [14], we conduct an ablation experiment to study it. Table 4 shows the comparison results of batch sizes from 64 to 4096. We use the same linear scaling rule—$base\_lr * Batch\_size * 128$ (the *base_lr* is 0.3)—for all experiments.

Table 4. The effect of batch size.

Batch Size/STS	STS12	STS13	STS14	STS15	Avg.
64	**56.35**	72.56	66.59	74.73	67.56
128	56.20	**72.77**	66.72	75.03	**67.68**
256	55.94	72.66	66.67	75.22	67.62
512 (ours)	55.79	72.72	**66.73**	**75.40**	67.66
1024	55.73	72.44	66.45	**75.40**	67.51

Table 4 reports the results of batch sizes from 64 to 1024. Different from previous conclusions, our model is insensitive to batch size. On the contrary, the performance is worse when the batch size increases to 1024, compared with the batch size of 512. In addition, a small batch size of 64 also achieves competitive performance. A reasonable explanation is that the computing of contrastive loss does not include negative examples.

5.2. Effect of Weight Decay

We find that the value of the weight decay influences performance dramatically. We conjecture that the perturbation of model weight can influence contrastive self-supervised training. Therefore, we perform an experiment to investigate it. The results are shown in Table 5.

The experimental results show that improper weight decay could make the model stop training early, resulting in underfitting and poor performance.

Table 5. The effect of weight decay.

Weight Decay/STS	STS12	STS13	STS14	STS15	Avg.
0.0001	55.04	70.79	65.55	73.36	66.19
0.001 (ours)	**55.79**	**72.72**	**66.73**	**75.40**	**67.66**
0.01	55.22	70.48	65.43	74.12	66.31
0.1	54.94	70.26	64.75	72.58	65.63

5.3. Effect of the LR of the Predictor

As mentioned by [16], the predictor with a constant LR (without decay) can obtain good image representation. Therefore, we design an experiment to verify whether the same settings can obtain good sentence representation. The results are shown in Table 6.

Table 6. The effect of the LR of the predictor. Decay: the LR of the predictor reduces with a cosine decay.

LR/STS	STS12	STS13	STS14	STS15	Avg.
Decay	54.64	70.58	65.33	72.92	65.87
0.08	55.22	70.17	65.17	73.10	65.92
0.2	**56.24**	72.22	66.53	75.25	67.56
0.5	56.14	72.59	66.73	75.36	**67.71**
1 (ours)	55.79	**72.72**	**66.73**	**75.40**	67.66

Experimental results show that a predictor with a constant LR can obtain better sentence representation compared with a decay LR. Specifically, as shown in Table 6 and Figure 4, the model will stop training (at the 9th epoch) when the LR of the predictor is small or reduced by a linear scaling rule. Additionally, the model needs a bigger learning rate (LR = 1) compared with vision tasks (LR = 0.1) to obtain better results. A possible explanation is that the predictor can adapt the latest representation. Therefore, it is not necessary to force the predictor to converge before the model is trained sufficiently [16].

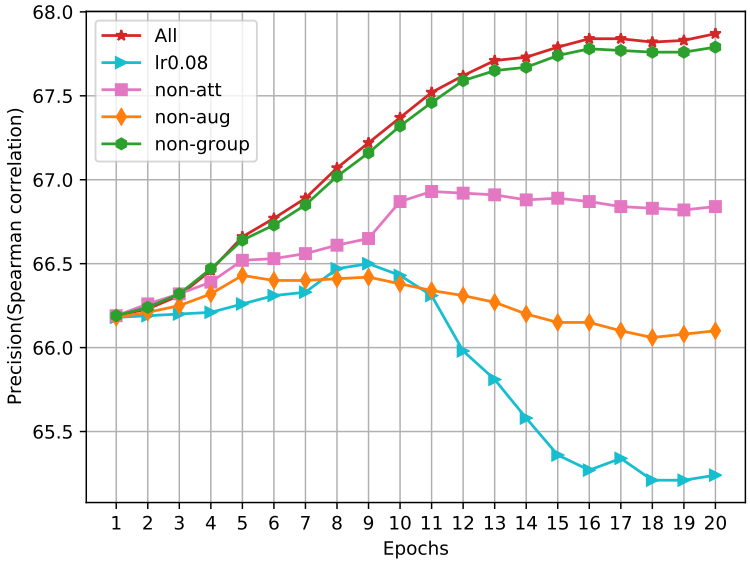

Figure 4. Comparison experiment results. "All" means that all methods we proposed are used. "lr0.08" means that the learning rate of the predictor is 0.08. "non-att": the self-attention we proposed is not used in whole training. "non-aug": no data augmentations are applied in training, i.e., the two channels of network receive the same input. "non-group": feature grouping is not adopted to make use of local information of features.

5.4. Effect of the SGD Momentum

In general, an optimizer with momentum can accelerate training because the update of the next step is based on the former steps. In other words, the gradient has a certain initial velocity (the network can remember the direction of gradient descent), which makes the network get rid of the local optima. In our proposed methods, the momentum is set to 0.9 before the warm-up epochs and 0.8 after the warm-up epochs. More details are shown in Table 7.

We observe that a small momentum will take more time to train the model and will not necessarily achieve the best performance. While a large momentum can save training time, the model can miss the optima resulting from a big step updating in the vicinity of the optimal point. Therefore, we set the momentum to (0.9, 0.8) to accelerate the training before warm-up epochs and to slow the updating step after the warm-up epochs, achieving better performance.

Table 7. The effect of the SGD momentum (Mot). (0.9,0.8) means that the momentum is 0.9 before warm-up and 0.8 after warm-up.

Momentum/STS	STS12	STS13	STS14	STS15	Avg.
0.8	55.47	72.45	66.50	74.84	67.32
0.9	55.27	71.96	66.27	75.18	67.17
0.99	54.76	70.86	65.69	73.93	66.24
(0.9,0.8) (ours)	**55.79**	**72.72**	**66.73**	**75.40**	**67.66**

5.5. Effect of the Warm-Up

In the training phase, the LR is linearly scaling, i.e., the LR linearly increases to the maximum and reduces to the minimum, which can make a model more robust. Given that the parameters of a model are randomly initialized, it is inappropriate to employ a large LR in the first few updates of training because the noise of the data may influence the performance. The comparison results are shown in Table 8.

Table 8. The effect of the warm-up. 1, 2, 3, 4, and 5 represent epochs that the LR starts to reduce.

Warm-Up/STS	STS12	STS13	STS14	STS15	Avg.
1	55.62	72.52	66.54	74.75	67.36
2	55.45	72.38	66.42	74.90	67.29
3	55.62	72.54	66.56	75.09	67.45
4	55.64	72.55	66.60	75.07	67.47
5 (ours)	**55.79**	**72.72**	**66.73**	**75.40**	**67.66**

Overall, the performances between the different warm-up epochs are comparable. However, a small warm-up can make the model stop early, especially for data with much noise.

5.6. Effect of the Region Size of TextCNN

The region size is a crucial parameter of TextCNN. Therefore, we design different region sizes to investigate their impacts. The results are shown in Table 9.

The experimental results show that the performance can be influenced dramatically by region size. Specifically, region size 1 is crucial for obtaining good results, observed from region size (1,2,3,4,5,6) and (2,3,4,5,6). A possible explanation is that region size 1 can enhance the representation of every word itself in a sentence without noise from other words. Furthermore, we can increase the region size to study it. The results show that, although large region sizes can obtain a better result compared with a small region sizes (1,1,1,2,3,4) on STS tasks, worse performance is obtained in transfer tasks (a large region size will reduce by 0.3 percentage points). We argue that a large region size could obtain more context information, but at the same time, much noise is also added into the representation.

Table 9. The effect of the region size of TextCNN.

Region Size/STS	STS12	STS13	STS14	STS15	Avg.
(1,2,3,4,5,6)	55.36	67.92	63.33	73.55	65.04
(2,3,4,5,6)	53.85	62.36	59.82	70.80	61.71
(1,1,1,2,3,4)	55.52	71.72	65.79	74.90	66.98
(1,1,1,4,5,6)	56.71	71.35	65.53	75.19	67.20
(1,1,1,1,1,1)	**57.81**	71.01	65.83	**75.62**	67.57
(1,1,1,6,15,20) (ours)	55.79	**72.72**	**66.73**	75.40	**67.66**
(1,1,1,20,30,40)	57.44	70.82	66.00	75.35	67.40

5.7. Effect of Data Augmentation

Data augmentation could affect the quality of positive samples used for contrastive learning, which directly influences the performance and robustness of the model. Consequently, we propose two hypotheses for discrete text data: (1) partial data augmentation could preserve more original semantic information; (2) continuous data augmentation could guarantee that there is no semantic gap in augmented data. Next, we conduct an experiment to verify it. The results are shown in Table 10 and Figure 4.

Table 10. The effect of data augmentation. No Aug.: no word vectors are subject to the data augmentation. Full Aug.: all of the word vectors of a sentence are augmented with our proposed four data augmentation strategies. Partial Aug.: word vectors are augmented by our proposed PWVA.

Augmentation/STS	STS12	STS13	STS14	STS15	Avg.
No Aug.	54.65	70.63	65.39	72.96	65.91
Full Aug.	55.31	70.51	65.81	73.49	66.28
Partial Aug. (ours)	**55.79**	**72.72**	**66.73**	**75.40**	**67.66**

As shown in experimental results, the continuous and PWVA improve the performance compared with No Aug. (from 65.91 to 67.66) and Full Aug. (from 66.28 to 67.66), which verifies our two hypotheses about data augmentation used in contrastive learning. In addition, the model can work well without data augmentation. A possible explanation is that unrecognizable words' random initialization can be regarded as a method of data augmentation, resulting in an improvement in stability and robustness.

5.8. Effect of the Size of Groups

Grouping the features of the projector and predictor can solve the issue of information loss caused by contrastive loss computing between high-dimensional vectors. Therefore, we conduct an experiment to study the effect of the size of the feature grouping. The results are shown in Table 11.

Table 11. The effect of the size of feature grouping. No grouping: feature grouping is not performed.

Grouping Size/STS	STS12	STS13	STS14	STS15	Avg.
No grouping	55.77	72.69	66.67	75.33	67.62
4	55.77	72.62	66.61	**75.50**	67.63
8	55.79	72.63	66.65	75.35	67.61
16 (ours)	**55.79**	**72.72**	**66.73**	75.40	**67.66**
32	55.73	72.45	66.57	75.39	67.54
128	55.75	72.63	66.59	75.26	67.56

Generally speaking, different grouping sizes can achieve comparable performance on STS tasks. Although the performance gap between feature grouping and no grouping is small, as observed in Figure 4, the stability and robustness of the model with feature grouping are better compared with no grouping. This verifies that the usage of local information

by feature grouping can help the model mine more information for contrastive learning to advance the performance of sentence representation slightly (from 66.66 to 66.75).

6. Conclusions and Future Work

Previous work used large pre-trained language models to perform sentence representation (such as BERT and RoBERT), but could not evaluate the performance of a lightweight model on sentence representation using contrastive learning. In this paper, we propose a lightweight model GCLSR to investigate the effectiveness of contrastive learning for sentence representation. GCLSR consists of continuous and partial data augmentation PWVA, self-attention, and grouped contrastive learning. GCLSR can obtain more original semantics from PWVA to produce high-quality positive samples. Self-attention can help GCLSR focus on informative words. Grouped contrastive learning can make use of more local information of features. The experimental results show that our proposed method of GCLSR can produce a meaningful sentence representation. Additionally, the findings of PWVA have practical implications. PWVA conducts data augmentation in a partial and continuous manner in the word embedding space. However, there are some limitations. For example, our proposed method is evaluated on a lightweight model, i.e., TextCNN, and achieves promising results, while the effectiveness of it on a large model is uncertain. In the future, we intend to combine contrastive learning with self-attention further. In addition, we will use our proposed method on a large pre-trained language model (such as BERT or GPT) to obtain better results regarding sentence representation.

Author Contributions: Conceptualization, Q.W.; methodology, Q.W.; software, Q.W. and W.Z.; validation, T.L.; formal analysis, W.Z.; investigation, Q.W.; resources, Q.W.; data curation, T.L.; writing—original draft preparation, Q.W.; writing—review and editing, Q.W.; visualization, W.Z. and D.P.; project administration, D.P.; funding acquisition, D.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research is financially supported by the Sichuan Science and Technology Planning Project (2021YFG0301, 2021YFG0317, 2023YFQ0020, 2023YFG0033, 2023ZHCG0016, 2022YFQ0014, 2022YFH0021), Chengdu Science and Technology Project (2023-XT00-00004-GX).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Our code and training data for GCLSR are accessed on 21 July 2023 by https://github.com/qianandfei/GCLSR.

Acknowledgments: The authors would like to thank Sichuan Science and Technology Planning Project (2021YFG0301, 2021YFG0317, 2023YFQ0020, 2023YFG0033, 2023ZHCG0016, 2022YFQ0014, 2022YFH0021), Chengdu Science and Technology Project (2023-XT00-00004-GX) for financial support.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhao, D.; Wang, J.; Lin, H.; Chu, Y.; Wang, Y.; Zhang, Y.; Yang, Z. Sentence representation with manifold learning for biomedical texts. *Knowl.-Based Syst.* **2021**, *218*, 106869. [CrossRef]
2. Li, B.; Zhou, H.; He, J.; Wang, M.; Yang, Y.; Li, L. On the sentence embeddings from pre-trained language models. *arXiv* **2020**, arXiv:2011.05864.
3. Logeswaran, L.; Lee, H. An efficient framework for learning sentence representations. *arXiv* **2018**, arXiv:1803.02893.
4. Kim, T.; Yoo, K.M.; Lee, S.g. Self-Guided Contrastive Learning for BERT Sentence Representations. *arXiv* **2021**, arXiv:2106.07345.
5. Zhang, D.; Li, S.W.; Xiao, W.; Zhu, H.; Nallapati, R.; Arnold, A.O.; Xiang, B. Pairwise supervised contrastive learning of sentence representations. *arXiv* **2021**, arXiv:2109.05424.
6. Ethayarajh, K. How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings. *arXiv* **2019**, arXiv:1909.00512.
7. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv* **2019**, arXiv:1910.10683.

8. Dong, L.; Yang, N.; Wang, W.; Wei, F.; Liu, X.; Wang, Y.; Gao, J.; Zhou, M.; Hon, H.W. Unified language model pre-training for natural language understanding and generation. *arXiv* **2019**, arXiv:1905.03197.
9. Wu, L.; Hu, J.; Teng, F.; Li, T.; Du, S. Text semantic matching with an enhanced sample building method based on contrastive learning. *Int. J. Mach. Learn. Cybern.* **2023**, *14*, 3105–3112. [CrossRef]
10. Ma, X.; Li, H.; Shi, J.; Zhang, Y.; Long, Z. Importance-aware contrastive learning via semantically augmented instances for unsupervised sentence embeddings. *Int. J. Mach. Learn. Cybern.* **2023**, *14*, 2979–2990. [CrossRef]
11. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
12. Liu, P.; Guo, Y.; Wang, F.; Li, G. Chinese named entity recognition: The state of the art. *Neurocomputing* **2022**, *473*, 37–53. [CrossRef]
13. Yu, P.; Weizhong, Q. Three-stage question answering model based on BERT. *J. Comput. Appl.* **2022**, *42*, 64.
14. Chen, T.; Kornblith, S.; Norouzi, M.; Hinton, G. A simple framework for contrastive learning of visual representations. In Proceedings of the International Conference on Machine Learning, Virtual, 13–18 July 2020; pp. 1597–1607.
15. He, K.; Fan, H.; Wu, Y.; Xie, S.; Girshick, R. Momentum contrast for unsupervised visual representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 9729–9738.
16. Chen, X.; He, K. Exploring simple siamese representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 15750–15758.
17. Grill, J.B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P.H.; Buchatskaya, E.; Doersch, C.; Pires, B.A.; Guo, Z.D.; Azar, M.G.; et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv* **2020**, arXiv:2006.07733.
18. Reimers, N.; Gurevych, I. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv* **2019**, arXiv:1908.10084.
19. Giorgi, J.M.; Nitski, O.; Bader, G.D.; Wang, B. Declutr: Deep contrastive learning for unsupervised textual representations. *arXiv* **2020**, arXiv:2006.03659.
20. Wu, Z.; Wang, S.; Gu, J.; Khabsa, M.; Sun, F.; Ma, H. Clear: Contrastive learning for sentence representation. *arXiv* **2020**, arXiv:2012.15466.
21. Gao, T.; Yao, X.; Chen, D. SimCSE: Simple Contrastive Learning of Sentence Embeddings. *arXiv* **2021**, arXiv:2104.08821.
22. Wang, Q.; Zhang, W.; Lei, T.; Cao, Y.; Peng, D.; Wang, X. CLSEP: Contrastive learning of sentence embedding with prompt. *Knowl.-Based Syst.* **2023**, *266*, 110381. [CrossRef]
23. Fang, H.; Wang, S.; Zhou, M.; Ding, J.; Xie, P. Cert: Contrastive self-supervised learning for language understanding. *arXiv* **2020**, arXiv:2005.12766.
24. Zhu, W.; Cheung, D. CMV-BERT: Contrastive multi-vocab pretraining of BERT. *arXiv* **2020**, arXiv:2012.14763.
25. Yan, Y.; Li, R.; Wang, S.; Zhang, F.; Wu, W.; Xu, W. ConSERT: A Contrastive Framework for Self-Supervised Sentence Representation Transfer. *arXiv* **2021**, arXiv:2105.11741.
26. Wei, J.; Zou, K. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv* **2019**, arXiv:1901.11196.
27. Wang, W.Y.; Yang, D. That's so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using# petpeeve tweets. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 2557–2563.
28. Guo, H.; Mao, Y.; Zhang, R. Augmenting data with mixup for sentence classification: An empirical study. *arXiv* **2019**, arXiv:1905.08941.
29. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv* **2014**, arXiv:1409.0473.
30. Uchaikin, V.V.; Zolotarev, V.M. *Chance and Stability: Stable Distributions and Their Applications*; Walter de Gruyter: Berlin, Germany, 2011.
31. Géron, A. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2022.
32. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
33. Kim, Y. Convolutional Neural Networks for Sentence Classification. *arXiv* **2014**, arXiv:1408.5882.
34. Ioffe, S.; Normalization, C.S.B. Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv* **2015**, arXiv:1502.03167.
35. Loshchilov, I.; Hutter, F. Sgdr: Stochastic gradient descent with warm restarts. *arXiv* **2016**, arXiv:1608.03983.
36. Conneau, A.; Kiela, D. Senteval: An evaluation toolkit for universal sentence representations. *arXiv* **2018**, arXiv:1803.05449.
37. Agirre, E.; Cer, D.; Diab, M.; Gonzalez-Agirre, A. Semeval-2012 task 6: A pilot on semantic textual similarity. In Proceedings of the 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), Montréal, Canada, 7–8 June 2012; pp. 385–393.
38. Agirre, E.; Cer, D.; Diab, M.; Gonzalez-Agirre, A.; Guo, W. * SEM 2013 shared task: Semantic textual similarity. In Proceedings of the Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity, Atlanta, GA, USA, 13–14 June 2013; pp. 32–43.
39. Agirre, E.; Banea, C.; Cardie, C.; Cer, D.; Diab, M.; Gonzalez-Agirre, A.; Guo, W.; Mihalcea, R.; Rigau, G.; Wiebe, J. Semeval-2014 task 10: Multilingual semantic textual similarity. In Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), Dublin, Ireland, 23–24 August 2014; pp. 81–91.

40. Agirre, E.; Banea, C.; Cardie, C.; Cer, D.; Diab, M.; Gonzalez-Agirre, A.; Guo, W.; Lopez-Gazpio, I.; Maritxalar, M.; Mihalcea, R.; et al. 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), Denver, CO, USA, 4–5 June 2015; pp. 252–263.
41. Agirre, E.; Banea, C.; Cer, D.; Diab, M.; Gonzalez Agirre, A.; Mihalcea, R.; Rigau Claramunt, G.; Wiebe, J. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In Proceedings of the SemEval-2016, 10th International Workshop on Semantic Evaluation, San Diego, CA, USA, 16–17 June 2016; pp. 497–511.
42. Cer, D.; Diab, M.; Agirre, E.; Lopez-Gazpio, I.; Specia, L. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv* **2017**, arXiv:1708.00055.
43. Marelli, M.; Menini, S.; Baroni, M.; Bentivogli, L.; Bernardi, R.; Zamparelli, R. A SICK cure for the evaluation of compositional distributional semantic models. In Proceedings of the LREC 2014, Reykjavik, Iceland, 26–31 May 2014; pp. 216–223.
44. Pang, B.; Lee, L. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. *arXiv* **2005**, arXiv:cs/0506075.
45. Hu, M.; Liu, B. Mining and summarizing customer reviews. In Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, WA, USA, 22–25 August 2004; pp. 168–177.
46. Pang, B.; Lee, L. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. *arXiv* **2004**, arXiv:cs/0409058.
47. Wiebe, J.; Wilson, T.; Cardie, C. Annotating expressions of opinions and emotions in language. *Lang. Resour. Eval.* **2005**, *39*, 165–210. [CrossRef]
48. Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C.D.; Ng, A.Y.; Potts, C. Recursive deep models for semantic compositionality over a sentiment treebank. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Seattle, WA, USA, 18–21 October 2013; pp. 1631–1642.
49. Voorhees, E.M.; Tice, D.M. Building a question answering test collection. In Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Athens, Greece, 24–28 July 2000; pp. 200–207.
50. Dolan, W.B.; Brockett, C. Automatically constructing a corpus of sentential paraphrases. In Proceedings of the Third International Workshop on Paraphrasing (IWP2005), Jeju Island, Republic of Korea, 4 October 2005.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.