3R: Enhancing Sentence Representation Learning via Redundant Representation Reduction

Longxuan Ma, Xiao Wu, Yuxin Huang*, Shengxiang Gao, Zhengtao Yu
kunming university of science and technology
Faculty of Information Engineering and Automation
lxma@kust.edu.cn wuxiao1@stu.kust.edu.cn huangyuxin2004@163.com
gaoshengxiang.yn@hotmail.com ztyu@hotmail.com

Abstract

Sentence representation learning (SRL) aims to learn sentence embeddings that conform to the semantic information of sentences. In recent years, fine-tuning methods based on pre-trained models and contrastive learning frameworks have significantly advanced the quality of sentence representations. However, within the semantic space of SRL models, both word embeddings and sentence representations derived from word embeddings exhibit substantial redundant information, which can adversely affect the precision of sentence representations. Existing approaches predominantly optimize training strategies to alleviate the redundancy problem, lacking fine-grained guidance on reducing redundant representations. This paper proposes a novel approach that dynamically identifies and reduces redundant information in a dimensional perspective, training the SRL model to redistribute semantics on different dimensions, and entailing better sentence representations. Extensive experiments across seven semantic text similarity benchmarks demonstrate the effectiveness and generality of the proposed method. A comprehensive analysis of the experimental results is conducted and the code/data will be released.

1 Introduction

Sentence representation learning (SRL) (Yan et al., 2021; Zhou et al., 2022) is a fundamental task that aims to learn sentence embeddings that benefit downstream tasks such as semantic similarity (Agirre et al., 2016; Cer et al., 2017), information retrieval (Thakur et al., 2021), and sentiment analysis (Bao et al., 2023).

Recently, a training paradigm based on pre-trained models and contrastive learning as a fine-tuning method has achieved significant success in SRL. Among these, SimCSE (Simple Contrastive Learning of Sentence Embeddings) (Gao et al., 2021) stands out as a representative work (Chen et al., 2020; Sun et al., 2022; Liu et al., 2024). It proposes a simple yet effective method for constructing positive examples, significantly enhancing the quality of sentence embeddings. Subsequently, numerous studies (Chuang et al., 2022; He et al., 2023; Zhuo et al., 2023; Nguyen et al., 2024; Xu et al., 2024; Zhu et al., 2024) have focused on improving the SimCSE method to learn more effective sentence representations, including using large language models (LLMs) to evaluate training data quality (Cheng et al., 2023a) or directly generate high-quality data (Wang et al., 2024a).

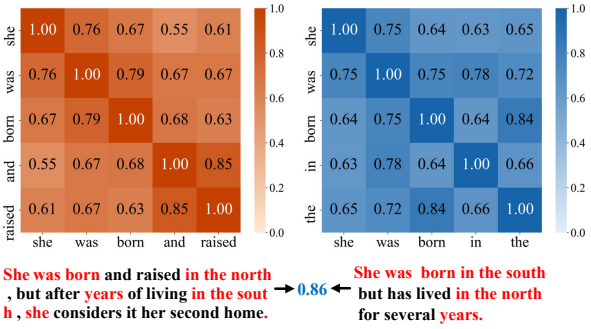

Figure 1: Word and sentence embedding redundancy.

Nevertheless, the contrastive SRL still faces certain challenges. **Firstly**, two sentences with significant semantic differences may still use the same words, with high-frequency words being the most common example, as shown in the bottom part of Figure 1. While some high-frequency words such as stop words play a crucial role in enhancing sentence coherence and semantic fluency, their contribution to the core semantics of the sentence is limited (Chen et al., 2022). High-frequency words add redundant encoded information, making it harder for SRL models to distinguish sentences with these overlapping words. **Secondly**, token embeddings learned by pre-trained models often exhibit redundant or ineffective information (Shi et al., 2022; Chen et al., 2023), leading to high similarity between tokens with different parts of speech and

*Corresponding author

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 31619–31632

meanings (with high-frequency words contributing to the majority). As shown in Figure 1, the cosine similarity between the embeddings of "was" and "born" reaches 0.79 in the upper left heat map, and the cosine similarity between "the" and "born" is 0.84 in the upper right heat map[1]. Sentence representation derived from token embeddings is influenced by the token-level redundant information (Tian et al., 2020), making it difficult for the SRL models to understand the overall semantics.

These two challenges bring unexpected redundant information (Shen et al., 2023), which hinders the contrastive SRL models from focusing on key semantic details and acquiring adequate discriminative knowledge (Chen et al., 2022, 2023). For instance, despite the semantic gap, the cosine similarity between two sentence representations reaches 0.86 in Figure 1. Current studies on contrastive SRL normally address the redundancy problem by adjusting training strategies. For example, Chen et al. (2022) proposes an information minimization-based contrastive learning method to learn the important information and drop the redundant information; Chen et al. (2023) utilizes hidden representations from intermediate layers as negative samples which the final sentence representations should be away from. However, these training strategies often lack fine-grained guidance to identify and reduce redundancy, hindering the model's ability to learn better sentence representations.

In this paper, we propose a **R**edundant **R**epresentation **R**eduction (3R) approach, which adopts an explicit signal to guide the reduction of redundant representations. The 3R method comprises three steps: (1) constructing a corpus-level redundant sentence embedding based on high-frequency words, (2) enabling the model to self-identify dimensions containing redundant information within each training batch, and (3) dynamically reducing redundant information for each training sample using the corpus-level redundant embedding and self-identified redundant dimensions. The 3R method helps the SRL model adjust the information distribution in different dimensions and enhances the ability of SRL models to concentrate on critical semantic information, thereby learning better sentence representations.

The 3R method offers several advantages: 1) it can be implemented with several lines of code; 2) the method is model-agnostic and it requires no modification to the core network architecture of SimCSE. Hence, it can be easily adopted to different contrastive learning-based representation learning frameworks; 3) experiments show that 3R can help the contrastive SRL model learn effective representations that improve downstream task performance. The contributions of this paper are:

- We propose a Redundant Representation Reduction (3R) method that dynamically identifies and reduces redundant information in dimensions, which helps the contrastive SRL model to focus on key semantic information and learn better sentence representation.

- Extensive experiments on standard semantic textual similarity (STS) tasks demonstrate that 3R: 1) outperform previous approaches that aim to improve SRL by reducing redundant information; 2) work together with previous methods to improve performance, demonstrating good generality. We provide a systematic analysis of the results. The code and data will be released on GitHub[2].

2 Related Work

2.1 Contrastive Learning

Recently, contrastive learning-based approaches have become the primary direction in SRL (Gao et al., 2021). Contrastive learning aims to pull representations of similar samples closer while pushing representations of dissimilar samples as far apart as possible. The objective of unsupervised contrastive learning is shown in Equation (1):

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

where $\mathbf{h}_i$ represents the embedding of sample x_i in the deep learning model, and $\mathbf{h}_i^+$ denotes the embedding of the positive example of x_i. $\mathbf{h}_j^+$ is the embedding of the examples within the same training batch, $j \in \{1, 2, ...N\}$. $\mathrm{sim}(\cdot)$ is the cosine similarity between two representations. τ is a temperature constant, which adjusts the influence of the similarity scores on the loss L_i. Building on the unsupervised framework, the objective of supervised contrastive learning introduces hard negative samples (Liu et al., 2025), as shown below:

[1]Computed with SimCSE(BERT-base).

[2]https://github.com/malongxuan/3R

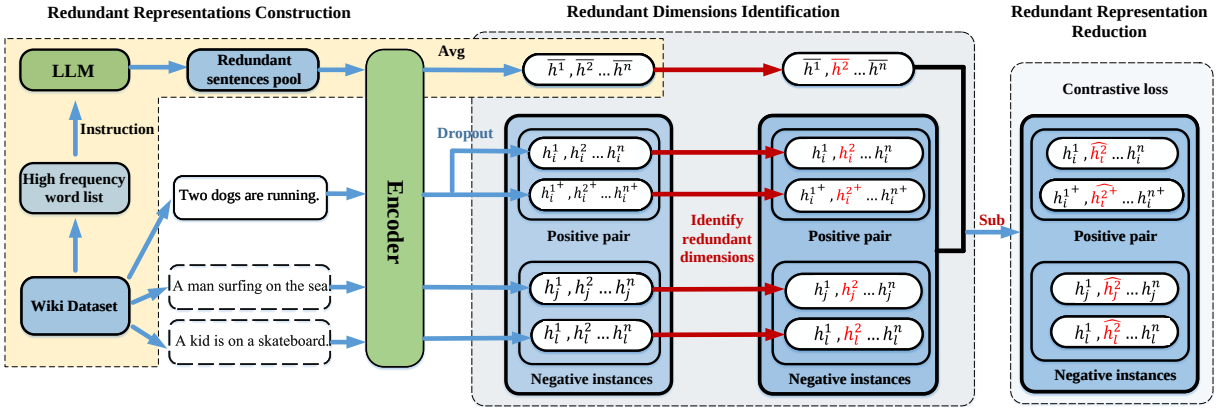

Figure 2: The proposed Redundant Representation Reduction (3R) Method.

$$-\log \frac{e^{\text{sim}(\mathbf{h}_i,\mathbf{h}_i^+)/\tau}}{\sum_{j=1}^{N}\left(e^{\text{sim}(\mathbf{h}_i,\mathbf{h}_j^+)/\tau}+e^{\text{sim}(\mathbf{h}_i,\mathbf{h}_j^-)/\tau}\right)}, \quad (2)$$

where $\mathbf{h}_j^-$ represents the embedding of the negative example x_j^- for the sample x_j in the deep learning model (Ma et al., 2022). *Our method aims to reduce redundant information of the sentence representation* ***h****, which can be adapted to both unsupervised and supervised contrastive learning.*

2.2 SimCSE and Its Improvements

To make contrastive SRL more effective, considerable research efforts have been paid to construct high-quality training examples. SimCSE (Gao et al., 2021) is a representative work that constructs positive pairs through the Dropout mechanism in neural networks. It feeds the same sentence into the model twice and uses the embeddings generated from these two passes as positive pairs in unsupervised contrastive learning. Wu et al. (2022) proposes to construct positive pairs through word repetition, which effectively alleviates the bias issue caused by the length similarity of positive pairs. Wang and Dou (2023) uses a rule-based method to construct semantically opposite but structurally identical sentences as negatives. Xu et al. (2023a) adopts an adversarial learning framework to construct both positive and negative pairs. Xu et al. (2023b) improves SRL by removing the Dropout noise in negative pairs. In recent years, some studies leverage the LLMs to select (Cheng et al., 2023a) or construct (Jiang et al., 2022; Cheng et al., 2023a; Wang et al., 2024a) high-quality training data, or directly use LLMs as the base model for contrastive SRL (Li and Li, 2024).

Another line of research aims to improve the quality of sentence representations by optimizing the contrastive learning objective, such as integrating semantic information (Tan et al., 2022), incorporating soft-prompt information (Ou and Xu, 2024), and introducing additional loss terms (Chuang et al., 2022; Lee, 2023).

Among the previous work, Chen et al. (2022), Shen et al. (2023) and Chen et al. (2023) are similar to ours that try to improve SRL by reducing redundant information. (Chen et al., 2022) introduces an additional loss function to guide the model in encoding less redundancy into sentence embeddings. Shen et al. (2023) proposes a post-processing method to subtract sentence-level and corpus-level redundant information in sentence embeddings. Chen et al. (2023) treats representations of sentences from intermediate layers of the model as additional negative examples and reduces the redundancy in sentence embeddings by increasing the distance to these negative examples. However, the previous work lacks fine-grained guidance (Ma et al., 2024) on allocating effective semantic information. *In contrast, 3R provides guidance for the contrastive SRL to dynamically identify and reduce redundant information from each dimension.*

3 The Proposed 3R Method

As shown in Figure 2, the 3R method consists of redundant representation construction, redundant dimension identification, and redundant representation reduction.

3.1 Redundant Representations Construction

Inspired by the corpus-level redundancy defined by Shen et al. (2023), to facilitate the model in identifying and autonomously mitigating the influence of redundant information, we start with constructing a set of redundant exemplars derived from high-frequency lexical items within the train-

ing corpus. This type of exemplar represents both global semantic statistical information and the role of a "weak" example, which **cannot** provide effective semantic information to distinguish between positive and negative examples. Reducing this ineffective semantic information in sentence representation can help the model better focus on key semantics that distinguish different examples.

Step (1). We count the word frequencies in the unsupervised training dataset of the SimCSE framework (Wiki dataset (Gao et al., 2021)). For example, the top 10 high-frequency words and their corresponding frequencies are: ["the" $(1,437,106)$, "of" $(678,338)$, "in" $(583,691)$, "and" $(568,586)$, "a" $(413,817)$, "to" $(408,457)$, "was" $(250,346)$, "is" $(186,236)$, "on" $(170,559)$, "as" $(169,463)$]. During the experiments, we select the top 300 most frequent words for the next step. Using corpus-level term frequency statistics provides a more robust and comprehensive representation of the semantic distribution within the corpus.

Step (2). We construct the redundant exemplars with the top 300 frequent words. We adopt an open-sourced LLM Deepseek-v3[3] for this task. During each call to the model, we randomly select 50 words from the 300 high-frequency word list and then use the chosen words to generate redundant exemplars[4]. The specific instruction for using the large language model is: "*Please use the words provided to generate 5 sentences with different meanings. The requirement is that most of the words in the sentences should come from the provided vocabulary, and the other words in the sentences should also be as common as possible. The sentence length should not exceed 32*[5]*. The words provided are: "and, was, ...".*"

The reason for constructing "high-redundancy" sentences, rather than directly using high-frequency words, is that the goal of contrastive training is to learn effective sentence representations. Directly using high-frequency words leads to "redundant" representations that lack sentence-level semantic information. In the experimental section, we will compare the effect of directly using high-frequency words to generate redundant representations (Please refer to section 5.3).

[3]https://github.com/deepseek-ai/DeepSeek-V3

[4]The experiments to choose 50 and 300 are shown in Appendix B and F.

[5]The maximum sentence truncation length of the SimCSE model is 32. To ensure the integrity of sentence semantics, we set 32 as the length threshold for sentence generation.

1: He has worked for this prestigious company in the city for several years and is highly regarded for his professionalism.
2: The group of students from the university were discussing the film until late into the night, trying to decide if it was the best they had seen this year.

Table 1: Two redundant sentence examples.

Step (3). Repeat step (2) until a sufficient number of sentences are obtained. We constructed sentences to ensure that they covered the top 300 most frequent words, and each frequent word was used at least twice. Finally, we had 64[6] sentences. Table 1 shows two examples of the constructed high-redundancy sentences. The underlined words are from the high-frequency word list.

The set of 64 sentences constructed in this section will serve as a candidate pool. We randomly select k sentences $(0 < k \leq 64)$ from this pool for each training batch. These sentences will be input to the encoder to get their embedding[7] $\bar{\mathbf{h}}_l, l = \{1, 2, ..., k\}$. The mean encoding result $\bar{\mathbf{h}}$ of the k sentences will be used as the representative redundant representation for each batch during training.

3.2 Redundant Dimensions Identification

After obtaining the redundant representation $\bar{\mathbf{h}}$, we designed a batch-wise redundant dimensions identification method. It helps the model to identify the redundant dimensions during training.

According to the fundamental principles of information theory, a system with higher uncertainty carries a greater amount of information (MacKay, 2003). Based on this theory, we compute the standard deviation of the sentence embeddings across each dimension for the data in the same batch. A smaller standard deviation indicates a lower variance in that dimension. Hence the dimension's contribution to distinguishing between different embeddings is minimal. The mean u^d and standard deviation σ^d of the d-th dimension are computed with Equation (3):

$$\sigma^d = \sqrt{\frac{\sum_{j=1}^{N}(h_j^d - u^d)^2}{N}}, \; u^d = \frac{1}{N}\sum_{j=1}^{N} h_j^d, \quad (3)$$

where N denotes the number of training data points

[6]The training batch size of the SimCSE model is 64. Please refer to section 5.5 for the related experiments.

[7]All sentence embeddings in the experiments are obtained with the encoding result of the [CLS] token in BERT (Devlin et al., 2019) or RoBERTa (Liu et al., 2019).

in the batch, $\mathbf{h}_j$ represents the j-th sentence embedding in the batch, $j \in \{1, 2, ..., N\}$. n is the hidden size of the sentence embedding. h_j^d is the value of the d-th dimension for the embedding of the j-th sentence, $d \in \{1, 2, ..., n\}$.

Then, we set up a learnable threshold c (an explicit signal) to decide which dimension is redundant. Specifically, the dimensions with a standard deviation smaller than c are defined as redundant for that training batch. We define S as the set of redundant dimensions selected. If $\sigma^d - c < 0$, $d \in S$, otherwise $d \notin S$.

3.3 Redundant Representations Reduction

After identifying the redundant dimensions, we propose a simple regularization method to reduce the redundant information in sentence embeddings. The regularized sentence embeddings can more accurately reflect the semantic distribution between sentences, which benefits the optimizing objective of contrastive learning.

As shown in Equations (4) and (5), the model discards redundant information in sentence embeddings by subtracting the redundant vector during the contrastive fine-tuning process. This redundancy reduction strategy is inspired by the work of (Shen et al., 2023), who used this method as a post-processing step to reduce redundant information in sentence embeddings. However, unlike their strategy of subtracting the overall vector, we only perform subtraction on the high-redundancy dimensions (as selected in Section 3.2).

$$L_i = -\log \frac{\exp\left(\text{sim}(\widehat{\mathbf{h}}_i, \widehat{\mathbf{h}}_i^+)/\tau\right)}{\sum_{j=1}^{N} \exp\left(\text{sim}(\widehat{\mathbf{h}}_i, \widehat{\mathbf{h}}_j^+)/\tau\right)}, \quad (4)$$

$$\begin{cases} \widehat{h}_x^d = h_x^d - \bar{h}^d, & \text{if } d \in S, \\ \widehat{h}_x^d = h_x^d, & \text{if } d \notin S, \end{cases} \quad (5)$$

where $\mathbf{h}_i = (h_i^1, h_i^2, \ldots, h_i^n)$ represents the i-th sentence embedding in the batch, $\bar{\mathbf{h}} = (\bar{h}^1, \bar{h}^2, \ldots, \bar{h}^n)$ denotes the constructed redundancy vector, n is the dimension of the vector, and $\bar{h}^d$ refers to the value at the d-th dimension of $\bar{\mathbf{h}}$. S is the set of selected redundant dimensions. $\mathbf{h}_x$ represents $\mathbf{h}_i$, $\mathbf{h}_j^+$, or $\mathbf{h}_i^+$. $\widehat{\mathbf{h}}$ is the reduced sentence representation we used for contrastive learning.

4 Experimental setting

4.1 Datasets and Evaluation Metrics

We evaluate the performance of sentence embeddings on the standard semantic textual similarity (STS) task, which includes seven sub-tasks[8]. Each sub-task requires the model to output a similarity score for a given sentence pair, with a score range from 0 to 5, where 0 indicates no semantic relevance and 5 indicates identical semantics. The evaluation metric of the STS task is the **Spearman correlation** between the predicted scores and human-annotated scores. We used the open-source code from (Gao et al., 2021) to compute the model's scores. The STS tasks are difficult for not only SRL models but also the state-of-the-art LLMs. Previous work (Wang et al., 2024a) shows that ChatGPT[9] equipped with in-context-learning (Dong et al., 2023) can only obtain 76.19 Spearman correlation score on this task, which is lower than many unsupervised methods based on BERT or RoBERTa (please refer to Table 2). Experiments on more backbone models and more downstream tasks are shown in Appendix D and E. All results are the average of five-times experiments.

Alignment and Uniformity are two metrics for evaluating the quality of the embedding space. Specifically, alignment measures the distance between positive pairs. A smaller alignment value indicates that semantically similar sentences are closer together in the vector space. Uniformity, on the other hand, evaluates the distribution of embeddings in the semantic space. A smaller uniformity value indicates a more uniform distribution of the vectors. Following (Reimers and Gurevych, 2019) and (Gao et al., 2021), who proposed the view that the primary objective of sentence embeddings is to cluster semantically similar sentences, we take alignment as the main results. In this study, we used the open-source code from (Wang and Isola, 2020) to compute the alignment and uniformity losses. The alignment loss is computed based on sentence pairs with similarity scores greater than 4 from the STS-B test set. The uniformity loss is computed with the entire STS-B test set.

4.2 Baselines

We compare with the following baselines.

Unsupervise SRL methods: (1) **SimCSE** (Gao et al., 2021) utilizes dropout for data augmentation in contrastive learning; (2) **InforMin-CL** (Chen et al., 2022) uses an additional loss

[8] STS12 (Agirre et al., 2012), STS13 (Agirre et al., 2013), STS14 (Agirre et al., 2014), STS15 (Agirre et al., 2015), STS16 (Agirre et al., 2016), STS-Benchmark (Cer et al., 2017), SICK-Relatedness (Marelli et al., 2014)

[9] https://openai.com/index/chatgpt/

Unsupervised Model	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg(Diff.)
SimCSE-BERT$_{\text{base}}$*	67.00	81.87	73.20	79.02	78.30	76.26	70.82	75.21
SimCSE-BERT$_{\text{base}}$*+ 3R	70.51	83.46	75.89	82.06	79.18	78.69	72.84	**77.52**(+2.31)
InforMin-CL-BERT$_{\text{base}}$*	66.64	82.10	73.32	78.15	77.33	75.70	71.30	74.93
InforMin-CL-BERT$_{\text{base}}$*+ 3R	71.52	81.41	75.11	81.84	78.19	79.25	73.34	**77.24**(+2.31)
RapAL-BERT$_{\text{base}}$	69.33	78.93	73.95	80.01	79.29	76.00	70.51	75.43
SSCL-SimCSE$_{\text{base}}$*	70.09	81.52	74.61	81.64	76.71	77.14	69.93	76.10
SSCL-SimCSE$_{\text{base}}$*+ 3R	71.79	83.62	76.51	83.54	78.61	79.54	71.83	**77.60**(+1.50)
SimCSE-Roberta$_{\text{base}}$*	69.18	81.71	72.50	81.10	80.31	79.68	69.99	76.35
SimCSE-Roberta$_{\text{base}}$*+ 3R	71.86	82.60	74.30	81.43	81.30	81.41	69.90	**77.54**(+1.19)
InforMin-CL-Roberta$_{\text{base}}$*	66.76	80.58	71.38	81.21	78.60	78.34	66.05	74.70
InforMin-CL-Roberta$_{\text{base}}$*+ 3R	67.79	82.81	74.33	82.99	79.53	81.71	71.89	**77.29**(+2.59)
LLM2Vec-LLaMA-2-7B*	70.20	81.76	73.83	81.37	78.32	76.75	70.79	76.15
LLM2Vec-LLaMA-2-7B*+ 3R	70.81	83.46	75.92	82.01	78.99	78.63	72.91	**77.53**(+1.38)
Supervised Model	**STS12**	**STS13**	**STS14**	**STS15**	**STS16**	**STS-B**	**SICK-R**	**Avg**(Diff.)
MultiCSRE-BERT$_{\text{base}}$*	72.48	82.75	75.94	82.51	80.07	81.89	77.38	79.00
MultiCSRE-BERT$_{\text{base}}$*+ 3R	73.05	81.14	76.23	83.32	80.55	82.43	77.82	**79.56**(+0.56)
SimCSE-BERT$_{\text{base}}$*	77.11	80.82	78.42	85.03	80.40	82.69	78.93	80.50
SimCSE-BERT$_{\text{base}}$*+ 3R	76.13	85.00	80.83	86.06	81.37	84.17	80.16	**81.96**(+1.46)
Claif-SimCSE-BERT$_{\text{base}}$*	76.89	79.59	79.06	85.93	81.01	83.68	79.08	80.75
Claif-SimCSE-BERT$_{\text{base}}$*+ 3R	76.06	84.76	80.99	86.10	81.41	81.81	79.60	**81.81**(+1.06)
SynCSE-scratch-BERT$_{\text{base}}$*	74.34	84.37	78.33	83.73	80.22	81.81	76.00	79.83
SynCSE-scratch-BERT-$_{\text{base}}$*+ 3R	76.65	83.26	79.52	84.81	81.02	83.82	79.70	**81.27**(+1.44)

Table 2: Experimental results on STS tasks. Results with $*$ are reproduced by us. The underlined scores are the best on each sub-task of each group. "Diff." means the improvement after using 3R method on the baselines.

function to incorporate less useless encodings into sentence embeddings; (3) **RapAL** (Shen et al., 2023) proposes a simple post-processing method to remove redundant information in sentence embeddings; (4) **SSCL** (Chen et al., 2023) reduces redundancy by trains the model away from similar intermediate layer representations; (5) **LLM2Vec** (BehnamGhader et al., 2024) enables bidirectional attention to decoder-only LLMs such as LLaMA-2-7B (Touvron et al., 2023) and then uses LLMs for unsupervised SRL.

Supervise SRL methods: (1) **SimCSE** (Gao et al., 2021); (2) **Claif** (Cheng et al., 2023b) uses an LLM to evaluate the quality of training data for supervised SRL; (3) **SynCSE-scratch** (Zhang et al., 2023) uses an LLM to construct training samples for supervised SRL; (4) **MultiCSR** (Wang et al., 2024b) uses an LLM for multiple stages generating and selecting high-quality sentences.

4.3 Training Details

The experiments were conducted with an RTX 4090 GPU. We followed the hyper-parameter settings from the previous works (Gao et al., 2021; Cheng et al., 2023b; Chen et al., 2023; BehnamGhader et al., 2024), training the unsupervised model with randomly sampled sentences from Wiki data, training the supervised model with MNLI and SNLI datasets, using the same pre-trained checkpoints of BERT (uncased) and RoBERTa (cased) for different methods.

5 Experimental Results and Analysis

In this section, we aim to answer the following questions: **1)** Does the 3R method outperform the previous methods that aim at reducing redundant information in contrastive SRL? (Section 5.1) **2)** Does the 3R method work together with the previous unsupervised methods (Section 5.1) and supervised methods (Section 5.2)? **3)** What are the advantages of the 3R method? (Section 5.2) **4)** How does each module contribute to the 3R method i.e. where do the gains come from? (Section 5.3) **5)** What can we learn from the case study? (Section 5.4) **6)** How is the hyper-parameter k decided? (Section 5.5) **7)** How is improvement reflected in the sentence representation space? (Appendix A)

5.1 Analysis of Unsupervised Methods

The upper half of Table 2 shows the experimental results with unsupervised methods. **Firstly**, the SimCSE+3R outperforms models (InforMin-CL, RagAL, and SSCL) that also aim at reducing redundant information. The results indicate that reducing redundancy from a fine-grained dimensional perspective may better mitigate the redundancy problem. **Secondly**, the proposed 3R method achieves performance im-

Unsupervised Model	STS12	STS13	STS14	STS15	STS16	STSB	SICK-R	Avg
SimCSE-BERT$_{\text{base}}$	67.00	81.87	73.20	79.02	78.30	76.26	70.82	75.21
SimCSE-BERT$_{\text{base}}$(dynamic mask)	68.12	82.53	74.39	80.73	77.77	77.41	72.43	76.20
SimCSE-BERT$_{\text{base}}$(overall subtraction)	72.09	82.71	74.94	80.96	78.23	78.07	70.88	76.84
SimCSE-BERT$_{\text{base}}$(token subtraction)	71.38	82.46	75.16	81.30	77.65	78.19	71.62	76.82
SimCSE-BERT$_{\text{base}}$(static identification)	69.31	81.85	74.88	80.78	78.30	77.31	71.60	76.29
SimCSE-BERT$_{\text{base}}$(3R)	70.51	83.46	75.89	82.06	79.18	78.69	72.84	**77.52**

Table 3: Different settings of the proposed 3R method. The underlined scores are the best on each sub-task.

provements on all BERT/RoBERTa/LLaMA models, showing a good generality on base models. **Thirdly**, InforMin-CL+3R outperforms InforMin-CL 2.31/2.59 on BERT/RoBERTa, respectively. SSCL-SimCSE+3R outperforms SSCL-SimCSE 1.5. The results demonstrate that our method can work with other redundant information reduction methods, further improving performance.

5.2 Analysis of Supervised Methods

The bottom half of Table 2 shows the experimental results on the STS tasks with supervised methods. The proposed 3R method achieves performance improvements on SimCSE, Claif-SimCSE, and SynCSE-scratch. SimCSE uses manually annotated training data. Claif-SimCSE adopts LLM to evaluate the quality of training data. SynCSE-scratch leverages LLM to construct training data. The effect of the 3R method is less pronounced under supervised conditions compared to unsupervised ones. One possible reason is that in supervised training batches, the semantic relationship between a sample and its hard negative pair is more complex. In such cases, the model can already learn better sentence representations through hard negative examples, so the improvement that reducing redundant information can bring is limited.

To sum up, the results in Table 2 show a good generality of the 3R method: **1)** the redundancy representation is an issue in both unsupervised and supervised training paradigms and the 3R method can mitigate the redundant representation problem in both training paradigms; **2)** the 3R method works for different types of data scenarios, whether automatically constructed or human annotated; **3)** the 3R method can work together with different redundancy reducing methods and different data augmentation methods.

5.3 Different Settings of 3R

Table 3 shows the experiments with different settings of 3R, which explain why 3R works and where the gains come from.

The "dynamic mask" setting does not perform Equation (5). Instead of subtracting the identified redundant dimensions of the redundant representation $\bar{\mathbf{h}}$, it directly sets the identified redundant dimension of $\mathbf{h}_x$ to 0. Hence, **this setting can be seen as removing the Redundant Representations Construction module of the 3R method**. This setting is better than the baseline model but worse than the original 3R. The results indicate that simply removing the identified redundant dimensions may also eliminate useful information. A more refined process for reducing the redundant information such as the 3R may be more appropriate.

The "overall subtraction" setting subtracts the constructed redundancy representation $\bar{\mathbf{h}}$ from all dimensions, instead of only subtracting the identified ones. **This setting removes the Redundant Dimensions Identification module**. The results are better than the baseline model, which shows that the constructed redundancy representation exemplifies the redundant information in the training data. Removing this redundancy helps to learn a better sentence representation. On the other hand, the "overall subtraction" setting is inferior to 3R, which means the learned threshold c helps to identify which dimension is worth more to reduce.

The "token subtraction" setting uses the 300 high-frequency words to obtain the representative redundant embedding. **It means we do not construct the redundant sentence pool.** We directly use the average embedding of the 300 high-frequency words as the redundant embedding $\bar{\mathbf{h}}$. Then we use the learnable c to decide which dimension is redundant and should reduced with Equation (5). We can observe that this setting can still improve the SimCSE model's performance, which means the $\bar{\mathbf{h}}$ derived from high-frequency words can also guide where the redundancy information is. However, this setting is not comparable to 3R, which means simply using high-frequency words could not provide enough sentence-level semantic information for contrastive SRL.

The "static identification" setting selects redun-

1. explosion at Venezuela refinery kills at least 39.	
2. Venezuela mourns oil refinery blast deaths.	
Human-annotated similarity score:	0.56
Similarity score from Claif / Claif+3R:	0.75 / 0.65

Table 4: Similarity score for a random case.

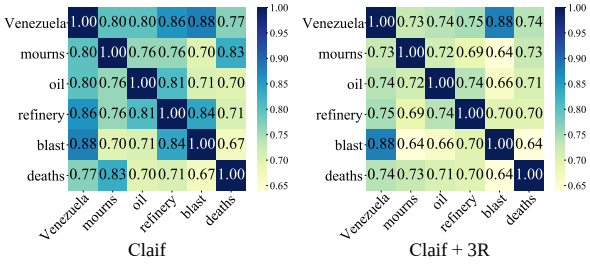

Figure 3: Token embedding similarity heat maps.

dant dimensions based on the 300 high-frequency words. Specifically, **we do not use the sentence embeddings in each batch to compute the standard deviation of each dimension**. Instead, we use the embedding of the 300 high-frequency words to calculate the standard deviation of each dimension. Then we use the learnable c to decide which dimension is redundant and should reduced with Equation (5). This means the redundant dimensions are the same among different batches. We can observe that this setting can improve the SimCSE model's performance, which means the high-frequency words can guide where the redundancy information is exhibited. However, this setting is not as good as 3R, which means dynamically determining redundant dimensions in each batch can better guide the model to learn the subtle semantic differences among different batches.

5.4 Case Study

We randomly select a sentence pair in the STS tasks for similarity study. As shown in Table 4, the two sentences have a manually annotated score of 0.56 (For the convenience of comparison, we convert the manually annotated scores between 0-5 to 0-1). The similarity score from Claif and Claif+3R is 0.75 and 0.65, respectively. The 3R method helps the contrastive SRL model to give a score closer to a human-annotated one.

We also show the word similarity of the second sentence "Venezuela mourns oil refinery blast deaths" in Figure 3. On one side, the 3R method makes the distinction between token embeddings with different parts of speech and meanings more pronounced. For example, the words "mourns" and "refinery" have a similarity score of 0.76 in

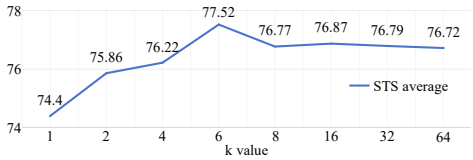

Figure 4: 3R method performance with different k.

Claif, while the score is 0.69 in Claif+3R. On the other side, the 3R method retains key semantic information while reducing the impact of redundant information. For instance, the words "Venezuela" and "blast" have a similarity score of 0.88 in both models. Although the similarity between dissimilar words has been reduced, there is still room for improvement. Further research to reduce unexpected redundant information is still needed.

5.5 The Hyper-parameter k

There is a hyper-parameter k in the method (section 3.1), which is the number of redundant exemplars randomly chosen from the redundant sentence pool for each training batch. Figure 4 shows the average results on the STS tasks with different k. The experiments are conducted with SimCSE-BERT$_{\text{base}}$, which has an average performance of 75.21 on STS. We can see that the performance of the model gradually increases as k grows. It surpasses SimCSE when k is 2, which means it takes multiple redundant sentences to obtain corpus-level semantics to guide the 3R method. It reaches its peak when k is 6, and then stabilizes as k continues to increase. When k is 6, the learned variable c (Section 3.2) will converge to 0.273. c determines whether a dimension should be reduced with Equation (5). More experiments about c is in Appendix C.

6 Conclusion

This study optimizes SRL by automatically detecting and reducing redundant information in dimensions. The proposed method helps models adjust the information distributions among dimensions and learn better sentence representations. Extensive experiments demonstrate the effectiveness and generality of the method. We present a systematic analysis to show why the proposed method works. Future work includes: 1) investigating more delicate control of the reduction process (For example, dividing redundant dimensions into multiple redundancy levels); 2) testing the 3R method in more downstream tasks that apply contrastive learning.

Limitations

Firstly, our method requires training the parameters of the SRL model. When applying the proposed 3R method to models with larger sizes (e.g. more than 7B), the training is expensive. Hence, the 3R method benefits smaller models (e.g. smaller than 1B), which still show great application value nowadays in specific tasks, domains, and scenarios. Secondly, the alignment and uniformity analysis show that the uniformity score can still improve, which indicates we can further refine the proposed 3R method to have a better representation space.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (No. U21B2027, 62266027, 62266028), the Yunnan Provincial Major Science and Technology Special Plan Projects (Grant Nos. 202402AG050007, 202302AD080003, 202303AP140008, 202502AD080016), the General Projects of Basic Research in Yunnan Province (Grant No. 202301AS070047, 202301AT070471, 202201BE070001-021).

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel M. Cer, Mona T. 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 interpretability. In *SemEval@NAACL-HLT*, pages 252–263. The Association for Computer Linguistics.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel M. Cer, Mona T. Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. Semeval-2014 task 10: Multilingual semantic textual similarity. In *SemEval@COLING*, pages 81–91. The Association for Computer Linguistics.

Eneko Agirre, Carmen Banea, Daniel M. Cer, Mona T. 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 *SemEval@NAACL-HLT*, pages 497–511. The Association for Computer Linguistics.

Eneko Agirre, Daniel M. Cer, Mona T. Diab, and Aitor Gonzalez-Agirre. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. In *SemEval@NAACL-HLT*, pages 385–393. The Association for Computer Linguistics.

Eneko Agirre, Daniel M. Cer, Mona T. Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *sem 2013 shared task: Semantic textual similarity. In **SEM@NAACL-HLT*, pages 32–43. Association for Computational Linguistics.

Xiaoyi Bao, Xiaotong Jiang, Zhongqing Wang, Yue Zhang, and Guodong Zhou. 2023. Opinion tree parsing for aspect-based sentiment analysis. In *ACL (Findings)*, pages 7971–7984.

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. Llm2vec: Large language models are secretly powerful text encoders. *CoRR*, abs/2404.05961.

Daniel M. Cer, Mona T. Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *SemEval@ACL*, pages 1–14. Association for Computational Linguistics.

Nuo Chen, Linjun Shou, Jian Pei, Ming Gong, Bowen Cao, Jianhui Chang, Jia Li, and Daxin Jiang. 2023. Alleviating over-smoothing for unsupervised sentence representation. In *ACL (1)*, pages 3552–3566.

Shaobin Chen, Jie Zhou, Yuling Sun, and Liang He. 2022. An information minimization based contrastive learning model for unsupervised sentence embeddings learning. In *COLING*, pages 4821–4831. International Committee on Computational Linguistics.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A simple framework for contrastive learning of visual representations. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

Qinyuan Cheng, Xiaogui Yang, Tianxiang Sun, Linyang Li, and Xipeng Qiu. 2023a. Improving contrastive learning of sentence embeddings from AI feedback. In *ACL (Findings)*, pages 11122–11138.

Qinyuan Cheng, Xiaogui Yang, Tianxiang Sun, Linyang Li, and Xipeng Qiu. 2023b. Improving contrastive learning of sentence embeddings from AI feedback. In *ACL (Findings)*, pages 11122–11138.

Yung-Sung Chuang, Rumen Dangovski, Hongyin Luo, Yang Zhang, Shiyu Chang, Marin Soljacic, Shang-Wen Li, Scott Yih, Yoon Kim, and James R.Glass. 2022. Diffcse:difference-based contrastive learning for sentence embeddings. In *NAACL-HLT*, pages 4207–4218. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*, pages 4171–4186. Association for Computational Linguistics.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *IWP@IJCNLP*. Asian Federation of Natural Language Processing.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. 2023. A survey for in-context learning. *CoRR*, abs/2301.00234.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *EMNLP (1)*, pages 6894–6910. Association for Computational Linguistics.

Hongliang He, Junlei Zhang, Zhenzhong Lan, and Yue Zhang. 2023. Instance smoothed contrastive learning for unsupervised sentence embedding. In *AAAI*, pages 12863–12871. AAAI Press.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *KDD*, pages 168–177. ACM.

Ting Jiang, Jian Jiao, Shaohan Huang, Zihan Zhang, Deqing Wang, Fuzhen Zhuang, Furu Wei, Haizhen Huang, Denvy Deng, and Qi Zhang. 2022. Promptbert: Improving BERT sentence embeddings with prompts. In *EMNLP*, pages 8826–8837. Association for Computational Linguistics.

Hyunjae Lee. 2023. D2CSE: difference-aware deep continuous prompts for contrastive sentence embeddings. *CoRR*, abs/2304.08991.

Xianming Li and Jing Li. 2024. Bellm: Backward dependency enhanced large language model for sentence embeddings. In *NAACL-HLT*, pages 792–804. Association for Computational Linguistics.

Yihong Liu, Chunlan Ma, Haotian Ye, and Hinrich Schütze. 2024. Translico: A contrastive learning framework to address the script barrier in multilingual pretrained language models. In *ACL (1)*, pages 2476–2499.

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. *CoRR*, abs/1907.11692.

Yuanxing Liu, Jiahuan Pei, Weinan Zhang, Ming Li, Wanxiang Che, and Maarten de Rijke. 2025. Augmentation with neighboring information for conversational recommendation. *ACM Trans. Inf. Syst.*, 43(3):62:1–62:49.

Longxuan Ma, Changxin Ke, Shuhan Zhou, Churui Sun, Wei-Nan Zhang, and Ting Liu. 2024. A self-verified method for exploring simile knowledge from pre-trained language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*, pages 1563–1576. ELRA and ICCL.

Longxuan Ma, Ziyu Zhuang, Weinan Zhang, Mingda Li, and Ting Liu. 2022. Self-eval: Self-supervised fine-grained dialogue evaluation. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 485–495. International Committee on Computational Linguistics.

David J. C. MacKay. 2003. *Information theory, inference, and learning algorithms*. Cambridge University Press.

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. In *LREC*, pages 216–223. European Language Resources Association (ELRA).

Cong-Duy Nguyen, Thong Nguyen, Xiaobao Wu, and Anh Tuan Luu. 2024. KDMCSE: knowledge distillation multimodal sentence embeddings with adaptive angular margin contrastive learning. In *NAACL-HLT*, pages 733–749. Association for Computational Linguistics.

Fangwei Ou and Jinan Xu. 2024. SKICSE: sentence knowable information prompted by llms improves contrastive sentence embeddings. In *NAACL(Short Papers)*, pages 141–146. Association for Computational Linguistics.

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

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *ACL*, pages 115–124. The Association for Computer Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert:sentence embeddings using siamese bert-networks. In *EMNLP/IJCNLP (1)*, pages 3980–3990. Association for Computational Linguistics.

Lingfeng Shen, Haiyun Jiang, Lemao Liu, and Shuming Shi. 2023. A simple and plug-and-play method for unsupervised sentence representation enhancement. *CoRR*, abs/2305.07824.

Han Shi, Jiahui Gao, Hang Xu, Xiaodan Liang, Zhenguo Li, Lingpeng Kong, Stephen M.S.Lee, and James T.Kwok. 2022. Revisiting over-smoothing in BERT from the perspective of graph. In *ICLR*. OpenReview.net.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*, pages 1631–1642. ACL.

Chenyu Sun, Hangwei Qian, and Chunyan Miao. 2022. CCLF: A contrastive-curiosity-driven learning framework for sample-efficient reinforcement learning. In *IJCAI*, pages 3444–3450. ijcai.org.

Haochen Tan, Wei Shao, Han Wu, Ke Yang, and Linqi Song. 2022. A sentence is worth 128 pseudo tokens: A semantic-aware contrastive learning framework for sentence embeddings. In *ACL (Findings)*, pages 246–256.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *NeurIPS Datasets and Benchmarks*.

Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. 2020. What makes for good views for contrastive learning? In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Ellen M. Voorhees and Dawn M. Tice. 2000. Building a question answering test collection. In *SIGIR*, pages 200–207. ACM.

Hao Wang and Yong Dou. 2023. SNCSE: contrastive learning for unsupervised sentence embedding with soft negative samples. In *ICIC (4)*, volume 14089 of *Lecture Notes in Computer Science*, pages 419–431. Springer.

Huiming Wang, Zhaodonghui Li, Liying Cheng, De Wen Soh, and Lidong Bing. 2024a. Large language models can contrastively refine their generation for better sentence representation learning. In *NAACL-HLT*, pages 7874–7891. Association for Computational Linguistics.

Huiming Wang, Zhaodonghui Li, Liying Cheng, De Wen Soh, and Lidong Bing. 2024b. Large language models can contrastively refine their generation for better sentence representation learning. In *NAACL-HLT*, pages 7874–7891. Association for Computational Linguistics.

Tongzhou Wang and Phillip Isola. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pages 9929–9939. PMLR.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Lang. Resour. Evaluation*, 39(2-3):165–210.

Xing Wu, Chaochen Gao, Liangjun Zang, Jizhong Han, Zhongyuan Wang, and Songlin Hu. 2022. Esimcse: Enhanced sample building method for contrastive learning of unsupervised sentence embedding. In *COLING*, pages 3898–3907. International Committee on Computational Linguistics.

Bo Xu, Shouang Wei, Luyi Cheng, Shizhou Huang, Hui Song, Ming Du, and Hongya Wang. 2023a. Hsimcse: Improving contrastive learning of unsupervised sentence representation with adversarial hard positives and dual hard negatives. In *IJCNN*, pages 1–8. IEEE.

Jiahao Xu, Wei Shao, Lihui Chen, and Lemao Liu. 2023b. Simcse++: Improving contrastive learning for sentence embeddings from two perspectives. In *EMNLP*, pages 12028–12040. Association for Computational Linguistics.

Jiahao Xu, Charlie Soh Zhanyi, Liwen Xu, and Lihui Chen. 2024. Blendcse: Blend contrastive learnings for sentence embeddings with rich semantics and transferability. *Expert Syst. Appl.*, 238(Part E):121909.

Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021. Consert: A contrastive framework for self-supervised sentence representation transfer. *CoRR*, abs/2105.11741.

Junlei Zhang, Zhenzhong Lan, and Junxian He. 2023. Contrastive learning of sentence embeddings from scratch. In *EMNLP*, pages 3916–3932. Association for Computational Linguistics.

Kun Zhou, Beichen Zhang, Wayne Xin Zhao, and Ji-Rong Wen. 2022. Debiased contrastive learning of unsupervised sentence representations. In *ACL(1)*, pages 6120–6130.

Dongsheng Zhu, Zhenyu Mao, Jinghui Lu, Rui Zhao, and Fei Tan. 2024. SDA: simple discrete augmentation for contrastive sentence representation learning. In *LREC/COLING*, pages 14459–14471. ELRA/ICCL.

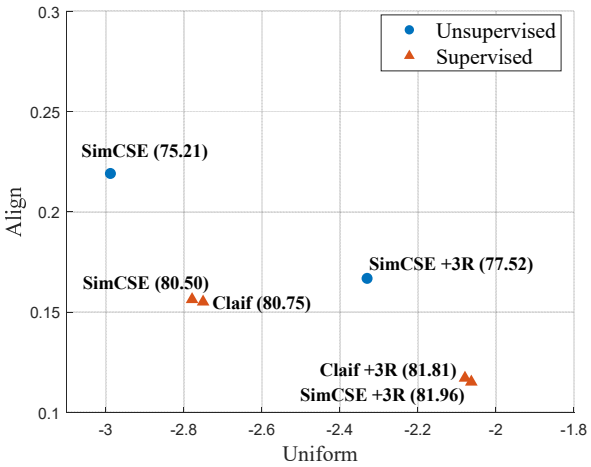

Figure 5: Align-Uniform Coordinate Plot.

Wenjie Zhuo, Yifan Sun, Xiaohan Wang, Linchao Zhu, and Yi Yang. 2023. Whitenedcse:whitening-based contrastive learning of sentence embeddings. In *ACL(1)*, pages 12135–12148.

A Alignment and Uniformity Analytics

In this section, we perform Alignment and Uniformity analysis of sentence embeddings with Figure 5. The alignment metric measures the distance between positive pairs. It drops after using the 3R method with both unsupervised and supervised models. The reason is that after reducing redundant information, the similarity of positive pairs decreases, which encourages the model to represent more granular semantic information in the positive pair embeddings. Thus, after redundancy reduction, semantically related sentences cluster more tightly in the embedding space. As we introduced in Section 4.1, a lower alignment score means a better model performance. Hence, the results show that the 3R method helps to learn better sentence representations.

The uniformity metric evaluates the distribution of embeddings in the semantic space. The models trained with 3R show a decrease in the uniformity metric compared to the baseline models. The reason is that the redundancy reduction operation removes some of the encodings on certain dimensions of the sentence embeddings, compressing the embedding space. The reduced degrees of freedom in the compressed embedding space cause sentence embeddings to cluster more easily. The results show a trade-off between alignment and uniformity. However, as pointed out by previous research (Reimers and Gurevych, 2019; Gao et al., 2021), the primary objective of sentence embeddings is to cluster semantically similar sentences. Hence, the results of the alignment-uniformity metric demonstrate the effectiveness of the 3R method in learning better sentence representations.

Index	100	101	102	103	104
Word	before	since	season	second	through
Frequency	14143	14053	14020	13874	13788
Changing rate	0.0048	0.0064	0.0024	**0.0105**	0.0062
Index	197	198	199	200	201
Word	another	former	members	York	any
Frequency	8143	8140	8060	8025	7978
Changing rate	0.0007	0.0004	**0.0099**	0.0044	0.0059
Index	297	298	299	300	301
Word	head	near	King	Road	off
Frequency	5811	5805	5795	5765	5761
Changing rate	0.0036	0.0010	0.0017	**0.0052**	0.0007

Table 5: Frequency statistics for choosing the top 300 frequency words.

B Experiments for choosing top 300 frequency words

Table 5 shows the experiments choosing the top 300 high-frequency words. Firstly, to have enough words to construct the required high-frequency sentence set, we empirically did not consider word lists smaller than 100. Then, we calculated the frequency changing rate with (T_n - T_{n+1}) / T_{n+1}, where T_n means the frequency of the n-th high-frequency word. After the calculation, we found that the changing rate exhibits local peaks at certain positions. For example, at 103, 176, 199, 300, 393, 472. Some of the statistics are as follows:

Based on the statistics, we conduct experiments on these local peaks to choose a number as the high-frequency word list. The experiments on unsupervised models are shown in Table 6.

The results show that the 300 setting is the best. The above statement and experiments are the rationale behind the value of 300. It is worth noting that all the settings (103, 176, 199, 300, 393, 472) are better than the baseline (75.21), which demonstrates that even if the optimal parameter 300 is not selected, the proposed method still works.

C Experiments for the learned parameter c

There is a learned parameter c in section 3.2 that is randomly initialized. We propose experiments with c set to different initial values (1.0, 0.5, 0.1, and 0.0), and the experimental results in one and a half epochs (Increasing by 0.1) are shown in Table 7. (0.0 is a very small number, such as 0.00001).

Model	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg
3R(length=103)	68.80	82.21	74.60	79.87	78.51	77.83	73.18	76.43
3R(length=176)	69.65	82.54	74.81	81.32	78.00	77.37	72.27	76.57
3R(length=199)	69.89	82.78	75.05	81.56	78.24	77.61	72.51	76.81
3R(length=300)	70.51	83.46	75.89	82.06	79.18	78.69	72.84	**77.52**
3R(length=393)	70.03	82.92	75.19	81.70	78.38	77.75	72.65	76.95
3R(length=472)	69.46	81.98	73.61	81.36	78.90	76.88	70.55	76.11

Table 6: Experiments for choosing the top 300 frequency words. The backbone model is SimCSE-BERT-base. The "length=" means the number of the top high-frequency words.

Epoch	0.1	0.2	0.3	0.4	0.5
c-initial=1.000	0.865	0.742	0.631	0.532	0.446
c-initial=0.500	0.443	0.365	0.319	0.298	0.285
c-initial=0.100	0.146	0.189	0.217	0.243	0.259
c-initial=0.000	0.032	0.067	0.103	0.142	0.176
Epoch	0.6	0.7	0.8	0.9	1.0
c-initial=1.000	0.373	0.314	0.273	0.273	0.273
c-initial=0.500	0.273	0.273	0.273	0.273	0.273
c-initial=0.100	0.267	0.273	0.273	0.273	0.273
c-initial=0.000	0.205	0.225	0.236	0.241	0.243
Epoch	1.1	1.2	1.3	1.4	1.5
c-initial=1.000	0.273	0.273	0.273	0.273	0.273
c-initial=0.500	0.273	0.273	0.273	0.273	0.273
c-initial=0.100	0.273	0.273	0.273	0.273	0.273
c-initial=0.000	0.255	0.265	0.273	0.273	0.273

Table 7: Experiments for the learned parameter c.

These experiments may provide more insight about our method.

We can see that c will converge to 0.273 with different initial values. It is worth noting that during the experiments in the paper, we set c to a random value between 0 and 1, and it also converges to 0.273.

To verify whether 0.273 is the optimal value. We conducted a comparison by manually setting the c value (which means c does not change or update during the training), and the experimental results with unsupervised models are shown in Table 8.

When c continues to decrease, the performance of the model will not improve, and 0.273 is the optimal value in our experiments. The best result here (77.46) is lower than the result (77.52) where c is automatically learned. It shows that automatically learn the threshold c helps the model to obtain a better generalization performance.

D Experiments on different back-bone models

Our experimental results on the BERT large and RoBERTa large models showed consistent trends with other experiments (the results of the Unsupervised models are shown in Table 9), indicating that using our method would improve the performance of the model.

For the LLMs, we present the experiments based on the LLAMA-7B model in Table 2 (i.e., LLM2vec) on page 6 of the paper, demonstrating that our method is also applicable to larger-scale models.

E Experiments on more downstream tasks

Following previous works, we evaluated our method on downstream tasks, and the results of the Unsupervised baselines and 3R method are shown in Table 10.

We evaluate our model performance on the following transfer tasks: 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), and MRPC (Dolan and Brockett, 2005).

Following previous work (Gao et al., 2021), we train a logistic regression classifier on top of the (frozen) sentence embeddings produced by different methods. The evaluation follows the default configuration of SentEval.

The results show that our method can also benefit the transfer tasks.

F Experiments for choosing hyper-parameter 50

In this section, we demonstrate how we choose the hyper-parameter 50. In the experiments, we randomly select N words from the high-frequency word list (103, 176, 199, 300, 393, 472) and then use the chosen words to generate redundant exemplars. We tried N = 40, 50, or 60. The experimental results are in Table 11. N = 30 or 70 were also tested but the results were much lower than the results of 40, 50, or 60. Table 11 shows that N = 50 is the best in all the settings (103, 176, 199, 300, 393, 472). Hence, we chose 50 as a hyper-parameter.

Model	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg
3R(c=0)	67.00	81.87	73.20	79.02	78.30	76.26	70.82	75.21
3R(c=0.1)	70.82	82.59	73.66	80.29	77.65	78.04	70.71	76.25
3R(c=0.2)	67.97	79.55	72.59	80.55	76.34	76.20	68.95	74.59
3R(c=0.25)	72.11	82.56	75.09	81.24	78.37	77.62	72.02	77.00
3R(c=0.273)	72.03	83.20	75.65	82.05	79.01	78.28	71.97	**77.46**
3R(c=0.3)	71.35	82.31	74.05	80.79	78.16	77.51	71.53	76.53

Table 8: Experiments for the learned parameter c. The backbone model is SimCSE-BERT-large.

Model	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg
SimCSE-BERT-large	69.44	83.71	75.74	83.90	78.66	78.53	73.70	77.67
SimCSE-BERT-large +3R	73.58	83.93	76.82	84.25	80.36	80.16	73.65	**78.96**
SimCSE-Roberta-large	72.18	83.15	75.13	84.11	81.11	81.66	71.01	78.34
SimCSE-Roberta-large +3R	74.36	83.72	76.68	84.53	82.01	82.21	72.56	**79.44**
MultiCSRE-Roberta$_{\text{base}}$	71.73	82.12	75.54	82.37	79.52	80.97	76.26	78.36
MultiCSRE-Roberta$_{\text{base}}$+3R	73.46	83.74	77.72	83.96	80.74	82.45	77.83	**79.99**

Table 9: Experiments with different backbone models

Model	MR	CR	SUBJ	MPQA	STS2	TREC	MRPC	Avg
SimCSE-BERT-base	81.11	85.56	94.20	89.17	85.56	86.40	74.14	85.16
SimCSE-BERT-base +3R	81.42	86.65	94.53	89.30	86.33	88.21	74.13	**85.80**
SimCSE-Roberta-base	80.57	86.62	92.27	86.61	85.72	83.20	73.97	84.14
SimCSE-Roberta-base +3R	80.73	86.87	93.42	87.13	86.01	85.37	74.08	**84.80**
SimCSE-BERT-large	85.05	89.48	95.01	89.29	90.44	88.80	74.20	87.47
SimCSE-BERT-large +3R	85.02	89.53	95.26	89.32	91.13	90.05	74.64	**87.85**
SimCSE-Roberta-large	82.59	87.47	93.18	88.44	86.66	91.00	76.29	86.52
SimCSE-Roberta-large +3R	83.54	87.65	93.14	88.76	87.02	90.87	76.25	**86.75**

Table 10: Experiments with different backbone models on downstream tasks.

Model	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg
3R(length=103,N=40)	68.62	82.13	74.36	79.69	78.33	77.56	73.01	76.24
3R(length=103,N=50)	68.80	82.21	74.60	79.87	78.51	77.83	73.18	**76.43**
3R(length=103,N=60)	68.73	82.08	74.52	79.75	78.44	77.71	73.07	76.33
3R(length=176,N=40)	69.54	82.42	74.73	81.21	77.83	77.21	72.10	76.43
3R(length=176,N=50)	69.65	82.54	74.81	81.32	78.00	77.37	72.27	**76.57**
3R(length=176,N=60)	69.58	82.49	74.73	81.28	77.91	77.32	72.23	76.51
3R(length=199,N=40)	69.73	81.58	74.87	81.36	78.08	77.42	72.34	76.48
3R(length=199,N=50)	69.89	82.78	75.05	81.56	78.24	77.61	72.51	**76.81**
3R(length=199,N=60)	69.83	82.69	74.94	81.47	78.15	77.54	72.46	76.73
3R(length=300,N=40)	70.48	83.40	75.61	81.58	79.06	78.11	72.77	77.29
3R(length=300,N=50)	70.51	83.46	75.89	82.06	79.18	78.69	72.84	**77.52**
3R(length=300,N=60)	72.03	83.20	75.65	82.05	79.01	78.28	71.97	77.46
3R(length=393,N=40)	69.78	82.63	74.89	81.45	78.07	77.44	72.40	76.67
3R(length=393,N=50)	70.03	82.92	75.19	81.70	78.38	77.75	72.65	**76.95**
3R(length=393,N=60)	69.84	82.73	74.94	81.51	78.10	77.52	72.43	76.72
3R(length=472,N=40)	69.29	81.77	73.44	81.15	78.72	76.68	70.36	75.92
3R(length=472,N=50)	69.46	81.98	73.61	81.36	78.90	76.88	70.55	**76.11**
3R(length=472,N=60)	69.38	81.89	73.53	81.28	78.81	76.76	70.46	76.02

Table 11: Experiments for choosing the 50 words. The backbone model is SimCSE-BERT-base.