Clustering-Aware Negative Sampling for Unsupervised Sentence Representation

Jinghao Deng[1], **Fanqi Wan**[1], **Tao Yang**[1], **Xiaojun Quan**[1]*, **Rui Wang**[2]
[1]School of Computer Science and Engineering, Sun Yat-sen University, China
[2]Vipshop (China) Co., Ltd., China
{dengjh27, wanfq, yangt225}@mail2.sysu.edu.cn,
quanxj3@mail.sysu.edu.cn, mars198356@hotmail.com

Abstract

Contrastive learning has been widely studied in sentence representation learning. However, earlier works mainly focus on the construction of positive examples, while in-batch samples are often simply treated as negative examples. This approach overlooks the importance of selecting appropriate negative examples, potentially leading to a scarcity of hard negatives and the inclusion of false negatives. To address these issues, we propose **ClusterNS** (**Cluster**ing-aware **N**egative **S**ampling), a novel method that incorporates cluster information into contrastive learning for unsupervised sentence representation learning. We apply a modified K-means clustering algorithm to supply hard negatives and recognize in-batch false negatives during training, aiming to solve the two issues in one unified framework. Experiments on semantic textual similarity (STS) tasks demonstrate that our proposed ClusterNS compares favorably with baselines in unsupervised sentence representation learning. Our code has been made publicly available.[1]

1 Introduction

Learning sentence representation is one of the fundamental tasks in natural language processing and has been widely studied (Kiros et al., 2015; Hill et al., 2016; Cer et al., 2018; Reimers and Gurevych, 2019). Reimers and Gurevych (2019) show that sentence embeddings produced by BERT (Devlin et al., 2019) are even worse than GloVe embeddings (Pennington et al., 2014), attracting more research on sentence representation with pre-trained language models (PLMs) (Devlin et al., 2019; Liu et al., 2019; Radford et al., 2019). Li et al. (2020a) and Ethayarajh (2019) further find out that PLM embeddings suffer from anisotropy, motivating more researchers to study this issue (Su et al., 2021; Gao et al., 2021). Besides, Gao et al. (2021) show that contrastive learning (CL) is able to bring significant improvement to sentence representation. As pointed out by Wang and Isola (2020), contrastive learning improves the uniformity and alignment of embeddings, thus mitigating the anisotropy issue.

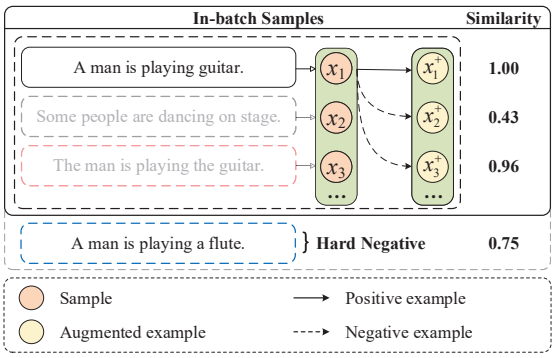


Figure 1: An example of in-batch negatives, a hard negative (in blue dotted box) and a false negative (in red dotted box). Cosine similarity is calculated with SimCSE (Gao et al., 2021). In-batch negatives may include false negatives, while lacking hard negatives.

Most previous works of constrastive learning concentrate on the construction of positive examples (Kim et al., 2021; Giorgi et al., 2021; Wu et al., 2020; Yan et al., 2021; Gao et al., 2021; Wu et al., 2022) and simply treat all other in-batch samples as negatives, which is sub-optimal. We show an example in Figure 1. In this work, we view sentences having higher similarity with the anchor sample as *hard negatives*, which means they are difficult to distinguish from positive samples. When all the negatives are sampled uniformly, the impact of hard negatives is ignored. In addition, various negative samples share different similarity values with the anchor sample and some may be incorrectly labeled (i.e., *false negatives*) and pushed away in the semantic space.

Recently quite a few researchers have demonstrated that hard negatives are important for contrastive learning (Zhang et al., 2022a; Kalantidis et al., 2020; Xuan et al., 2020). However, it is not

*Corresponding authors
[1]https://github.com/djz233/ClusterNS

Findings of the Association for Computational Linguistics: ACL 2023, pages 8713–8729

trivial to obtain enough hard negatives through sampling in the unsupervised learning setting. Admittedly, they can be obtained through retrieval (Wang et al., 2022b) or fine-grained data augmentation (Wang et al., 2022a), but the processes are usually time-consuming. Incorrectly pushing away false negatives in the semantic space is another problem in unsupervised learning scenarios, because all negatives are treated equally. In fact, in-batch negatives are quite diverse in terms of similarity values with the anchor samples. Therefore, false negatives do exist in the batches and auxiliary models may be required to identify them (Zhou et al., 2022). In sum, we view these issues as the major obstacles to further improve the performance of contrastive learning in unsupervised scenarios.

Since the issues mentioned above have a close connection with similarity, reasonable differentiation of negatives based on similarity is the key. In the meanwhile, clustering is a natural and simple way of grouping samples into various clusters without supervision. Therefore, in this paper, we propose a new negative sampling method called **ClusterNS** for unsupervised sentence embedding learning, which combines clustering with contrastive learning. Specifically, for each mini-batch during training, we cluster them with the K-means algorithm (Hartigan and Wong, 1979), and for each sample, we select its nearest neighboring centroid (cluster center) as the hard negative. Then we treat other sentences belonging to the same cluster as false negatives. Instead of directly taking them as positive samples, we use the Bidirectional Margin Loss to constrain them. Since continuously updating sentence embeddings and the large size of the training dataset pose efficiency challenges for the clustering, we modify the K-means clustering to make it more suitable for training unsupervised sentence representation.

Overall, our proposed negative sampling approach is simple and easy to be plugged into existing methods, boosting the performance. For example, we improve SimCSE and PromptBERT in $\mathrm{RoBERTa}_{\mathrm{base}}$ by 1.41/0.59, and in $\mathrm{BERT}_{\mathrm{large}}$ by 0.78/0.88 respectively. The main contributions of this paper are summarized as follows:

- We propose a novel method for unsupervised sentence representation learning, leveraging clustering to solve hard negative and false negative problems in one unified framework.
- We modify K-means clustering for unsupervised sentence representation, making it more efficient and achieve better results.
- Experiments on STS tasks demonstrate our evident improvement to baselines and we reach 79.74 for $\mathrm{RoBERTa}_{\mathrm{base}}$, the best result with this model.

2 Related Works

2.1 Contrastive Learning

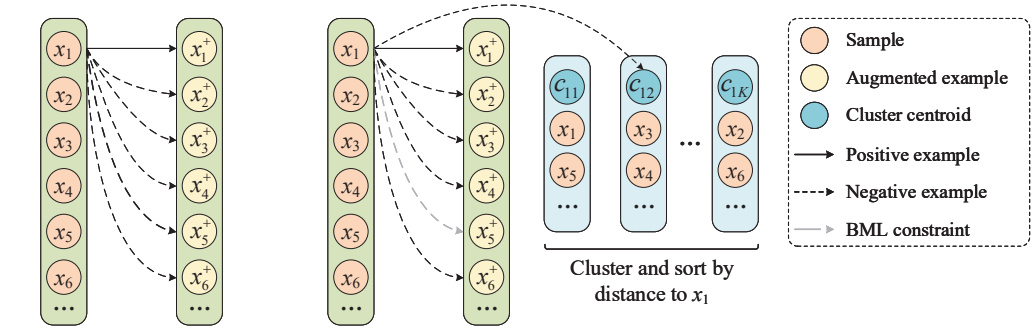
Contrastive learning is a widely-used method in sentence representation learning. Early works focus on positive examples, and have raised various kinds of effective data augmentations (Giorgi et al., 2021; Wu et al., 2020; Yan et al., 2021; Gao et al., 2021). Following these works, Wu et al. (2022) improve positive construction based on Gao et al. (2021). Zhou et al. (2022) improve the uniformity of negative. Besides, Zhang et al. (2022b) modify the objective function and Chuang et al. (2022) introduce Replaced Token Detection task (Clark et al., 2020), reaching higher performance.

2.2 Negative Sampling

In-batch negative sampling is a common strategy in unsupervised contrastive learning, which may have limitations as we mentioned above. To fix these issues, Zhang et al. (2022a) and Kalantidis et al. (2020) synthesize hard negatives by mixing positives with in-batch negatives. Wang et al. (2022a) utilize dependency parsing to create the negation of original sentences as soft negatives. Following Jiang et al. (2022) who use different prompt templates as positive, Zeng et al. (2022) derive negatives from the negation of the templates. The two methods create negatives with fixed templates and rules, thus may introduce bias. Chuang et al. (2020) design a debiased contrastive objective that corrects the false negatives without true labels. Zhou et al. (2022) use a trained model to distinguish false negatives, which results in addition module comparing with our method.

2.3 Neural Clustering

Clustering methods have been extended to deep learning and used for unsupervised representation learning (Xie et al., 2016; Yang et al., 2017; Caron et al., 2018; Li et al., 2020b; Zhang et al., 2021b). Prototypical Network (Snell et al., 2017), a variety of clustering, is widely used in few-shot learning (Cui et al., 2022; Ding et al., 2020; Gao et al., 2019). Several works have combined clustering



(a) Contrastive Learning Framework

(b) ClusterNS Framework

Figure 2: Illustrations of the original contrastive learning and our ClusterNS frameworks. Comparing with the naive framework, for the sample x_1 in the mini-batch, we provide the nearest neighboring centroid c_{12} as the hard negative, and regard the samples in the same cluster (such as x_5) as false negatives and constrain them by the BML loss.

with contrastive learning (Li et al., 2020b; Caron et al., 2020; Zhang et al., 2021b; Wang et al., 2021). Among them, Li et al. (2020b) argue that clustering encodes high-level semantics, which can augment instance-wise contrastive learning.

3 Methods

3.1 Preliminaries

Our clustering-based negative sampling method for unsupervised sentence representation can be easily integrated with contrastive learning approaches like SimCSE (Gao et al., 2021) or PromptBERT (Jiang et al., 2022). An illustration of ClusterNS and the original contrastive learning framework is shown in Figure 2. For a sentence x_i in one mini-batch $\{x_i\}_{i=1}^{N}$ (N samples in each mini-batch), SimCSE uses Dropout (Srivastava et al., 2014) and PromptBERT uses different prompt-based templates to obtain its positive example x_i^+. Then they treat the other samples in the mini-batch as "default" negatives and apply the *InfoNCE loss* (Oord et al., 2018) in Eq. (1), where τ is the temperature coefficient.

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

3.2 Boosting Negative Sampling

Our main contribution of this work is to improve the negative sampling method with clustering. To be more specific, we combine clustering with contrastive learning in the training process, recognizing false negatives in the mini-batch and providing additional hard negatives based on the clustering result. The clustering procedure will be introduced in Section 3.3 in detail and for the moment, we assume the samples in each mini-batch have been properly clustered.

Supposing that there are K centroids $c = \{c_i\}_{i=1}^{K}$ after clustering, standing for K clusters $C = \{C_i\}_{i=1}^{K}$. For a sample x_i in the mini-batch, we sort the clusters $C = [C_{i1},C_{i2},...,C_{iK}]$ and centroids $c = [c_{i1},c_{i2},...,c_{iK}]$ by their cosine similarity $cos(x_i, c_{ij})$ with x_i. In this case, c_{i1} and c_{iK} are the nearest and farthest centroids to x_i, respectively. Therefore, x_i is the most similar to c_{i1} and belongs to cluster C_{i1}. We define the set $x_i^* = \{x_{ij}^*\}_{j=1}^{count(C_{i1})}$, whose elements belong to C_{i1}, the same cluster as x_i.

Hard Negatives Zhang et al. (2022a) show that hard negatives bring stronger gradient signals, which are helpful for further training. The critical question is how to discover or even produce such negatives. In our method, the introduced centroids c can be viewed as hard negative candidates. We get rough groups in mini-batch after clustering and sorting by similarity. For the sample x_i, we pick the centroid c_{i2} as its hard negative. The reason is that c_{i2} gets the highest similarity with x_i among all the centroids (except for c_{i1} which x_i belongs to) while having a different cluster.

In this way, all the samples have proper centroids as their hard negatives, and the training objective $\mathcal{L}_{cl}$ is as follows:

$$\mathcal{L}_{cl} = -\log \frac{e^{sim(x_i,x_i^+)/\tau}}{\sum_{j=1}^{N}\left(e^{sim(x_i,x_j^+)/\tau} + \mu e^{sim(x_i,x_j^-)/\tau}\right)} \quad (2)$$

where x_j^- is the hard negative corresponding to x_j, μ is the weight of the hard negative. Note that c_{i1} is more similar to x_i compared with c_{i2}, which is another candidate for the hard negative. We have compared the different choices in the ablation study described in Section 4.4.

False Negatives For sample x_i, we aim to 1) recognize the false negatives in the mini-batch and 2) prevent them from being pushed away incorrectly in the semantic space. For the former, we treat elements in x_i^* as false negatives, since they belong to the same cluster and share higher similarity with x_i. For the latter, it is unreliable to directly use them as positives, since the labels are missing under the unsupervised setting. However, the different similarity between the anchor sample and others can be summarized intuitively as the following Eq. (3):

$$cos(x_i, x_i^-) \leq cos(x_i, x_{ij}^*) \leq cos(x_i, x_i^+) \quad (3)$$

where $x_{ij}^* \in x_i^*$. False negatives have higher similarity with the anchor than normal negatives while lower similarity than the positives. Inspired by Wang et al. (2022a), we introduce the bidirectional margin loss (BML) to model the similarity between the false negative candidates and the anchor:

$$\Delta_{x_i} = cos(x_i, x_i^*) - cos(x_i, x_i^+) \quad (4)$$

$$\mathcal{L}_{bml} = ReLU(\Delta_{x_i}+\alpha)+ReLU(-\Delta_{x_i}-\beta) \quad (5)$$

BML loss aims to limit $cos(x_i, x_i^*)$ in an appropriate range by limiting Δ_{x_i} in the interval $[-\beta, -\alpha]$. Accordingly, we find the potential false negatives in the mini-batch and treat them differently. Combining Eq. (2) and Eq. (5), we obtain the final training objective function as follows:

$$\mathcal{L} = \mathcal{L}_{cl} + \lambda \mathcal{L}_{bml} \quad (6)$$

where λ is a hyperparameter.

3.3 In-Batch Clustering

K-means clustering is the base method we use, while we need to overcome computational challenges during the training process. It is very inefficient to cluster the large training corpus. However, we need to do clustering frequently due to the continuously updating embeddings. Therefore, we design the training process with clustering in Algorithm 1. Briefly speaking, we use cosine similarity as the distance metric, cluster the mini-batch and update the centroids with momentum at each step.

Algorithm 1 Training with Clustering.

Input: Model parameters: θ; Training dataset: $\mathcal{D}$; Total update steps: T; Warm-up steps: S

1: **for** $t = 1$ to T **do**
2: Get the sentence embeddings $\{x_i\}_{i=1}^N$ for each mini-batch
3: **if** t == S **then**
4: Initialize centroids c with mini-batch samples heuristically
5: **end if**
6: **if** $t > S$ **then**
7: Update centroids c with $\{x_i\}_{i=1}^N$
8: Provide centroids as hard negatives $\{x_i^-\}_{i=1}^N$
9: Calculate $\mathcal{L}_{bml}$ for false negatives
10: **end if**
11: Calculate $\mathcal{L}_{cl}$
12: Loss backward and optimize θ
13: **end for**

Centroids Initialization We show the initialization in line 3–5 in Algorithm 1. The clustering is not performed at the beginning, since high initial similarity of embeddings harms the performance. Instead, we start clustering a few steps after the training starts, being similar to the warm-up process. When initializing, as line 4 shows, we select K samples as initial centroids heuristically: each centroid to be selected should be the least similar to last centroid.

Clustering and Updating We now describe line 7 in detail. First, we assign each sample into the cluster whose centroid have the highest cosine similarity with the sample. After clustering finishes, we calculate a new centroid embedding for each cluster by averaging embeddings of all samples in the cluster with Eq. (7), and then update the centroid in the momentum style with Eq. (8):

$$\widetilde{x_i} = \frac{1}{N_i} \sum_{x_j \in C_i} x_j \quad (7)$$

$$c_i = (1-\gamma)c_i + \gamma \widetilde{x_i} \quad (8)$$

where γ is the momentum hyperparameter and N_i indicates the number of elements in cluster C_i. Finally, based on the clustering results, we calculate the loss and optimize the model step by step (in line 9–12).

The method can be integrated with other contrastive learning models, maintaining high efficiency.

Models	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
Non-Prompt models								
GloVe embeddings	55.14	70.66	59.73	68.25	63.66	58.02	53.76	61.32
$\text{BERT}_{\text{base}}$ embeddings	39.70	59.38	49.67	66.03	66.19	53.87	62.06	56.70
$\text{BERT}_{\text{base}}$-flow	58.40	67.10	60.85	75.16	71.22	68.66	64.47	66.55
$\text{BERT}_{\text{base}}$-whitening	57.83	66.90	60.90	75.08	71.31	68.24	63.73	66.28
SimCSE-$\text{BERT}_{\text{base}}$	68.40	82.41	74.38	80.91	78.56	76.85	**72.23**	76.25
*ClusterNS-$\text{BERT}_{\text{base}}$	**69.93**	**83.57**	**76.00**	**82.44**	**80.01**	**78.85**	72.03	**77.55**
$\text{RoBERTa}_{\text{base}}$ embeddings	32.11	56.33	45.22	61.34	61.98	54.53	62.03	53.36
$\text{RoBERTa}_{\text{base}}$-whitening	46.99	63.24	57.23	71.36	68.99	61.36	62.91	61.73
SimCSE-$\text{RoBERTa}_{\text{base}}$	70.16	81.77	73.24	81.36	80.65	80.22	68.56	76.57
ESimCSE-$\text{RoBERTa}_{\text{base}}$	69.90	82.50	74.68	83.19	80.30	80.99	**70.54**	77.44
DCLR-$\text{RoBERTa}_{\text{base}}$	70.01	83.08	75.09	**83.66**	81.06	81.86	70.33	77.87
*ClusterNS-$\text{RoBERTa}_{\text{base}}$	**71.17**	**83.53**	**75.29**	82.47	**82.25**	**81.95**	69.22	**77.98**
SimCSE-$\text{BERT}_{\text{large}}$	70.88	84.16	76.43	84.50	79.76	79.26	73.88	78.41
MixCSE-$\text{BERT}_{\text{large}}$	**72.55**	84.32	76.69	84.31	79.67	79.90	74.07	78.80
DCLR-$\text{BERT}_{\text{large}}$	71.87	84.83	77.37	**84.70**	**79.81**	79.55	74.19	78.90
*ClusterNS-$\text{BERT}_{\text{large}}$	71.64	**85.97**	**77.74**	83.48	79.68	**80.80**	**75.02**	**79.19**
Prompt-based models								
PromptBERT$_{\text{base}}$	71.56	84.58	76.98	84.47	**80.60**	**81.60**	69.87	78.54
*ClusterNS-$\text{BERT}_{\text{base}}$	**72.92**	84.86	77.38	**84.52**	80.23	81.58	69.53	78.72
ConPVP-$\text{BERT}_{\text{base}}$	71.72	**84.95**	**77.68**	83.64	79.76	80.82	73.38	78.85
SNCSE-$\text{BERT}_{\text{base}}$	70.67	84.79	76.99	83.69	80.51	81.35	**74.77**	**78.97**
PromptRoBERTa$_{\text{base}}$	73.94	84.74	77.28	**84.99**	81.74	81.88	69.50	79.15
ConPVP-$\text{RoBERTa}_{\text{base}}$	73.20	83.22	76.24	83.37	81.49	82.18	**74.59**	79.18
SNCSE-$\text{RoBERTa}_{\text{base}}$	70.62	84.42	77.24	84.85	81.49	83.07	72.92	79.23
*ClusterNS-$\text{RoBERTa}_{\text{base}}$	**74.02**	**85.12**	**77.96**	84.47	**82.84**	**83.28**	70.47	**79.74**
PromptBERT$_{\text{large}}$	73.29	86.39	77.90	85.18	79.97	81.92	71.26	79.42
ConPVP-$\text{BERT}_{\text{large}}$	72.63	86.68	78.14	85.50	80.13	82.18	74.79	80.01
SNCSE-$\text{BERT}_{\text{large}}$	71.94	86.66	**78.84**	**85.74**	80.72	82.29	**75.11**	80.19
*ClusterNS-$\text{BERT}_{\text{large}}$	**73.99**	**87.53**	78.82	85.47	**80.84**	**82.85**	72.59	**80.30**

Table 1: Overall Results on STS tasks of Spearman's correlation coefficient. All baseline results are from original or relative papers. We use symbol * to mark our models. Best results are highlighted in bold.

4 Experiments

4.1 Evaluation Setup

Our experiments are conducted on 7 semantic textual similarity (STS) tasks (Agirre et al., 2012, 2013, 2014, 2015, 2016; Cer et al., 2017; Marelli et al., 2014) and the models are evaluated with the SentEval Toolkit (Conneau and Kiela, 2018). We take the Spearman's correlation coefficient as the metric and follow the Gao et al. (2021)'s aggregation method of results.

4.2 Implementation Details

Our code is implemented in Pytorch and Huggingface Transformers. The experiments are run on a single 32G Nvidia Tesla V100 GPU or four 24G Nvidia RTX3090 GPUs. Our models are based on SimCSE (Gao et al., 2021) and PromptBERT (Jiang et al., 2022), and named as *Non-Prompt* ClusterNS and *Prompt-based* ClusterNS, respectively. We use BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) as pre-trained language models for evaluation, with training for 1 epoch and evaluating each 125 steps on the STS-B development set. We also apply early stopping to avoid overfitting. Hyperparameter settings and more training details are listed in Appendix A.

4.3 Main Results

We present the experiment results in Table 1. We compare with four types of models totally, 1) vanilla embeddings of Glove, BERT and RoBERTa models, we report their results provided by Gao et al. (2021). 2) Baseline models: BERT-flow (Li et al., 2020a), BERT-whitening (Su et al., 2021), SimCSE (Gao et al., 2021) and Prompt-

BERT (Jiang et al., 2022). 3) SimCSE-based models: MixCSE (Zhang et al., 2022a), DCLR (Zhou et al., 2022) and ESimCSE (Wu et al., 2022). 4) PromptBERT-based models: ConPVP (Zeng et al., 2022) and SNCSE (Wang et al., 2022a). We compare SimCSE-based models with *Non-Prompt* ClusterNS, and PromptBERT-based models with *Prompt-based* ClusterNS, respectively. In this way, identical representation of sentence embeddings is guaranteed for a fair comparison.

Our conclusions are as follows, comparing with two baseline models, SimCSE and PromptBERT, all ClusterNS models achieve higher performance, indicating their effectiveness and the importance of negative sampling. For non-prompt models, ClusterNS surpasses MixCSE and DCLR in $\text{BERT}_{\text{large}}$, and for prompt-based models, ClusterNS also surpasses ConPVP and SNCSE in $\text{BERT}_{\text{large}}$ and $\text{RoBERTa}_{\text{base}}$. All these models improve negative samples through sampling or construction, demonstrating our models' strong competitiveness. At last, Prompt-based ClusterNS achieves the state-of-the-art performance of 79.74, which is the best result for models with $\text{RoBERTa}_{\text{base}}$.

4.4 Ablation Study

Our proposed method focuses on two issues, producing hard negatives and processing false negatives. To verify the contributions, we conduct the ablation studies by removing each of the two components on test sets of the STS tasks, with *Non-prompt* BERT and RoBERTa models. As we mentioned in Section 3.2, we also replace hard negatives with the most similar centroids to verify our choice of hard negatives (named *repl. harder negative*), and replace both centroids for hard and false negatives with random clusters to verify our choice of cluster centroids (named *repl. random clusters*). The results are in Table 2.

Models	$\text{BERT}_{\text{base}}$	$\text{RoBERTa}_{\text{base}}$
ClusterNS	77.55	77.98
w/o false negative	76.99(-0.56)	77.83(-0.15)
w/o hard negative	76.03(-1.52)	77.22(-0.76)
repl. harder negative	76.97(-0.58)	77.84(-0.14)
repl. random clusters	76.77(-0.78)	77.85(-0.13)
SimCSE	76.25	76.57

Table 2: Ablation results of our methods (*Non-prompt* Models) on the test set of STS tasks.

We observe from Table 2 that removing either component or replacing any part of models lead to inferior performance: 1) Providing hard negatives yields more improvement, since we create high similarity sample leveraging clustering. 2) Processing false negatives solely (without hard negatives) even further harm the performance, indicating that providing virtual hard negatives is much easier than distinguishing real false negatives. 3) Replacing hard negative with most similar centroids also degrades the performance. Since they belong to the identical cluster, the candidate hard negatives could be actually positive samples. And 4) random clusters are also worse, indicating that the selection of clusters does matter. We discuss more hyperparameter settings in Appendix E.

5 Analysis

To obtain more insights about how clustering helps the training process, we visualize the variation of diverse sentence pairs similarity during training after clustering initialization in the *Non-Prompt* ClusterNS-$\text{RoBERTa}_{\text{base}}$ model, and analyze the results in detail.

5.1 In-Batch Similarity

We visualize the average similarity of positive, in-batch negative and hard negative sentence pairs in Figure 3. We observe that similarity of in-batch negative drops rapidly as training progresses, indicating that in-batch negatives are difficult to provide gradient signal. The hard negatives provided by our method maintain higher similarity, which properly handles the issue. Also notice that the similarity of hard negatives is still much smaller than positive pairs, which avoids confusing the model.

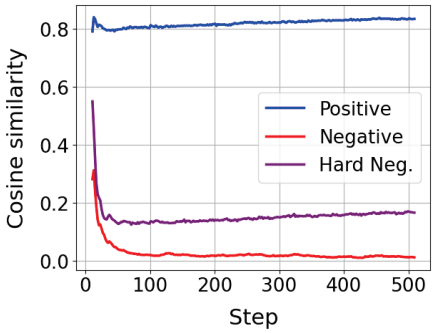


Figure 3: Variation for similarity of positive, in-batch negative and hard negative (Hard Neg.) pairs.

5.2 Clustering Similarity

Furthermore, we also visualize the similarity related to clustering. In Figure 4, we show the average similarity of sample-nearest centroid pairs,

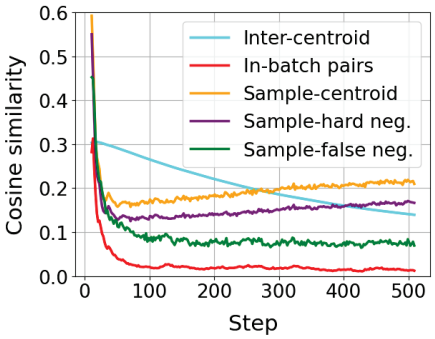


Figure 4: Variation for similarity of sample-nearest centroid pairs (Sample-centroid), Sample-hard negative pairs, Sample-false negative pairs (Intra-cluster member pairs), Inter-centroid pairs and In-batch negative pairs.

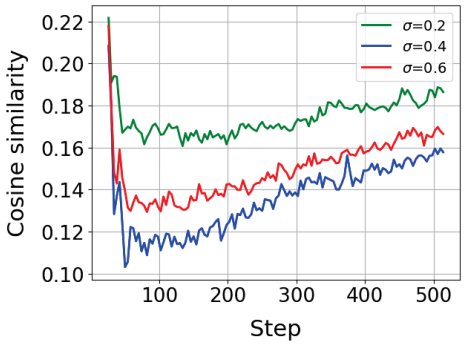


Figure 5: Varition for average similarity of hard negative pairs in different threshold σ

σ	0.2	0.4	0.6
Similarity	0.1814	0.1572	0.1442
Avg. STS	77.02	**77.98**	77.30
σ wo. L_{bml}	0.2	0.4	0.6
Similarity	0.1797	0.1574	0.1453
Avg. STS	77.43	77.83	77.46

Table 3: The average results of STS test sets in different threshold σ with\without L_{bml}, similarity means the average similarity of hard negative pairs at the last checkpoints.

sample-hard negative pairs (second nearest centroids, same as hard negative sentence pairs in Figure 3), inter-centroid pairs, intra-cluster member pairs (same as false negative pairs) and in-batch negative pairs. First, similarity of inter-centroid pairs decreases during training, demonstrating that clusters representing diverse semantics slowly scatter. Second, false negative pairs get much higher similarity than in-batch negatives, which indicates the importance of recognizing them and the necessity of treating them differently. At last, sample-nearest centroid pairs and sample-hard negative pairs maintain high similarity, demonstrating the stability of clustering during the training process.

To answer the question what is a *good* hard negative, we experiment with different similarity levels. We define a symbol σ, the average similarity threshold of in-batch sentence pairs when the centroids initialize. Since the similarity of hard negative pairs depends on σ, we adjust the similarity level with various σ settings.

We show the results in Figure 5 and Table 3. As we set the threshold σ smaller, clustering begins later and hard negatives gets larger similarity (with the anchor sample), meaning that starting clustering too early leads to less optimal hard negative candidates. The best performance is achieved at $\sigma = 0.4$, the middle similarity level, verifying the finding in our ablation study, i.e., hard negatives are not the most similar samples.

In Figure 4, similarity of false negative pairs is much smaller comparing with positive pairs, which shows the distinction between positive and false negative samples. False negatives are usually regarded as positive samples in supervised learning, while it is difficult to recognize precisely in the unsupervised setting. We argue that false negatives retrieved by our methods share similar topics with the anchors, leading to higher similarity than "normal" negatives and lower similarity than positives. We use Eq. (5) to constrain false negatives based on this hypothesis. We also do case studies and experiments to approve it in Appendix C.

To verify our choice of the BML loss, we implement experiments on different processing strategies of false negatives. We compare BML loss with two common strategies: use false negatives as positives and mask all the false negatives. Results in Table 4 demonstrate the superiority of the BML loss.

Models	Avg. STS
ClusterNS	77.98
w/o BML loss	77.83
Mask all false negatives	77.40
Use as positives	42.33

Table 4: Comparison of different false negative processing on STS test sets. *w/o BML loss* is the same as *w/o false negative* in Table 2.

6 Clustering Evaluation

We also evaluate the quality of sentence embedding through clustering. We first use the DBpedia dataset (Brümmer et al., 2016), an ontology classification dataset extracted from Wikipedia and consists of 14 classes in total. We implement K-means clustering (K=14) on the sentence embeddings of

Models	RoBERTa	SimCSE	ClusterNS
AMI	0.6926	0.7078	0.7355

Table 5: AMI score for K-means clustering (K=14) on DBpedia dataset. We use *Non-Prompt* ClusterNS for comparison. Higher values are better.

DBpedia, and take the adjusted mutual information (AMI) score as the evaluation metric following Li et al. (2020b). The results in Table 5 show that both sentence embedding models improve the AMI score, indicating that the cluster performance is positively correlated with the quality of sentence embeddings. ClusterNS achieves a higher AMI score than SimCSE, verifying its effectiveness.

Furthermore, we follow Zhang et al. (2021a) to conduct a more comprehensive evaluation of the short text clustering on 8 datasets [2], including AgNews (AG) (Zhang and LeCun, 2015), Biomedical (Bio) (Xu et al., 2017), SearchSnippets (SS) (Phan et al., 2008), StackOverflow (SO) (Xu et al., 2017), GoogleNews (G-T, G-S, G-TS) (Yin and Wang, 2016) and Tweet (Yin and Wang, 2016). We perform K-means clustering on the sentence embeddings and take the clustering accuracy as the evaluation metric. Results are shown in Table 6. Our ClusterNS models achieve higher performance than SimCSE in both two models, with an overall improvement of 4.34 in $\text{BERT}_{\text{base}}$. Both main experiments and two clustering evaluations show the improvement of our method to the baseline, and verify the effectiveness of improved negative sampling. More details about evaluation metrics are shown in Appendix D.

7 Alignment and Uniformity

To investigate how ClusterNS improves the sentence embedding, we conduct further analyses on two widely used metrics in contrastive learning proposed by Wang and Isola (2020), *alignment* and *uniformity*. Alignment measures the expected distance between the embeddings of positive pairs:

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

And uniformity measures the expected distance between the embeddings of all sentence pairs:

$$\mathcal{L}_{uniform} \triangleq \log \underset{(x,y)\sim p_{\text{data}}}{\mathbb{E}} e^{-2\|f(x)-f(y)\|^2} \quad (10)$$

[2]https://github.com/rashadulrakib/short-text-clustering-enhancement

Both metrics are better when the numbers are lower. We use the STS-B dataset to calculate the alignment and uniformity, and consider the sentence pairs with score higher than 4 as positive pairs. We show the alignment and uniformity of different models in Figure 6, along with the average STS test results. We observe that ClusterNS strikes a balance between alignment and uniformity, improving the weaker metric at the expense of the stronger one to reach a better balance. For the non-prompt models, SimCSE has great uniformity but weaker alignment compared to vanilla BERT and RoBERTa. ClusterNS optimizes the alignment. On the other hand, Prompt-based ClusterNS optimizes the uniformity since PromptRoBERTa performs the opposite of SimCSE. Besides, RoBERTa may suffer server anisotropy than BERT, meaning that sentence embeddings are squeezed in a more crowded part of the semantic space. Therefore, RoBERTa and PromptRoBERTa-untuned have extreme low value of alignment, but poor uniformity.

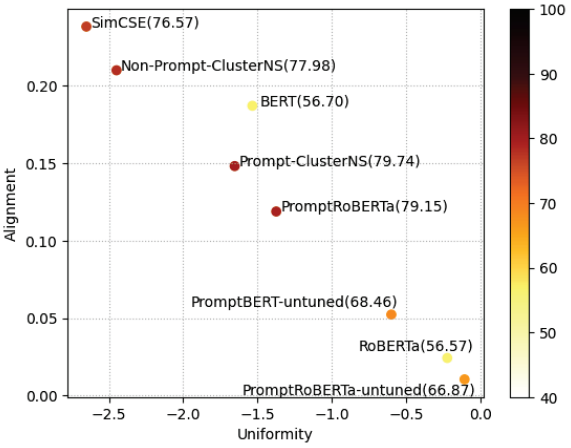


Figure 6: Alignment and uniformity for different sentence embedding models on the STS-B dataset. *untuned* means the models are not fine-tuned. We mainly use $\text{RoBERTa}_{\text{base}}$ models. Lower values are better.

8 Conclusion

In this paper, we propose ClusterNS, a novel approach that focuses on improving the negative sampling for contrastive learning in unsupervised sentence representation learning. We integrate clustering into the training process and use the clustering results to generate additional hard negatives and identify false negatives for each sample. We also use a bidirectional margin loss to constrain the false negatives. Our experiments on STS tasks show improvements over baseline models and demonstrate the effectiveness of ClusterNS. Through this work,

Models	AG	Bio	Go-S	G-T	G-TS	SS	SO	Tweet	Avg.
SimCSE-$\text{BERT}_{\text{base}}$	74.46	35.64	59.01	57.92	64.18	67.09	50.78	**54.71**	57.97
ClusterNS-$\text{BERT}_{\text{base}}$	**77.38**	**37.29**	**61.69**	**59.37**	**66.47**	**69.65**	**72.92**	53.71	**62.31**
SimCSE-$\text{RoBERTa}_{\text{base}}$	**69.71**	**37.35**	**60.89**	57.66	65.05	46.90	69.00	**51.89**	57.31
ClusterNS-$\text{RoBERTa}_{\text{base}}$	65.00	36.38	58.58	**57.88**	**65.54**	**52.55**	**74.38**	51.63	**57.74**

Table 6: Clustering accuracy on short text clustering datasets. We use *Non-Prompt* ClusterNS for comparison and evaluate on $\text{BERT}_{\text{base}}$ and $\text{RoBERTa}_{\text{base}}$. We reproduce all baseline results based on provided checkpoints. Best results are highlighted in bold.

we demonstrate that it is valuable to pay more attention to negative sampling when applying contrastive learning for sentence representation.

Acknowledgements

We appreciate the anonymous reviewers for their valuable comments. We thank Zhaoyang Wang for his support. This work was supported by the National Natural Science Foundation of China (No. 62176270), the Guangdong Basic and Applied Basic Research Foundation (No. 2023A1515012832), and the Program for Guangdong Introducing Innovative and Entrepreneurial Teams (No. 2017ZT07X355).

Limitations

Our work has two limitations. First, we update the cluster centroids at each step during training, which requires a large mini-batch to maintain clustering accuracy and consumes more GPU memory. Second, our method still may not identify false negatives accurately, as we use the training model for coarse-grained clustering rather than a well-trained model. We leave the improvement of memory consumption and further improving false negative discrimination for the future.

Ethics Statement

All datasets used in our work are from public sources, which do not consist private information. We strictly followed the data usage policy. Any research based on our work must sign an ethical statement and ensure that they do not infer user privacy from it.

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Iñigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. 2015. SemEval-2015 task 2: Semantic textual similarity, English, Spanish and pilot on interpretability. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 252–263, Denver, Colorado. Association for Computational Linguistics.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. SemEval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 81–91, Dublin, Ireland. Association for Computational Linguistics.

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. SemEval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 497–511, San Diego, California. Association for Computational Linguistics.

Eneko Agirre, Johan Bos, Mona Diab, Suresh Manandhar, Yuval Marton, and Deniz Yuret, editors. 2012. **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)*. Association for Computational Linguistics, Montréal, Canada.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *SEM 2013 shared task: Semantic textual similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 32–43, Atlanta, Georgia, USA. Association for Computational Linguistics.

Martin Brümmer, Milan Dojchinovski, and Sebastian Hellmann. 2016. DBpedia abstracts: A large-scale, open, multilingual NLP training corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3339–3343, Portorož, Slovenia. European Language Resources Association (ELRA).

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. 2018. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pages 132–149.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. 2020. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33:9912–9924.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium. Association for Computational Linguistics.

Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka. 2020. Debiased contrastive learning. *Advances in Neural Information Processing Systems*, 33:8765–8775.

Yung-Sung Chuang, Rumen Dangovski, Hongyin Luo, Yang Zhang, Shiyu Chang, Marin Soljacic, Shang-Wen Li, Scott Yih, Yoon Kim, and James Glass. 2022. DiffCSE: Difference-based contrastive learning for sentence embeddings. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4207–4218, Seattle, United States. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

Alexis Conneau and Douwe Kiela. 2018. SentEval: An evaluation toolkit for universal sentence representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Ganqu Cui, Shengding Hu, Ning Ding, Longtao Huang, and Zhiyuan Liu. 2022. Prototypical verbalizer for prompt-based few-shot tuning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7014–7024, Dublin, Ireland. 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 *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ning Ding, Xiaobin Wang, Yao Fu, Guangwei Xu, Rui Wang, Pengjun Xie, Ying Shen, Fei Huang, Hai-Tao Zheng, and Rui Zhang. 2020. Prototypical representation learning for relation extraction. In *International Conference on Learning Representations*.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65, Hong Kong, China. Association for Computational Linguistics.

Tianyu Gao, Xu Han, Zhiyuan Liu, and Maosong Sun. 2019. Hybrid attention-based prototypical networks for noisy few-shot relation classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6407–6414.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

John Giorgi, Osvald Nitski, Bo Wang, and Gary Bader. 2021. DeCLUTR: Deep contrastive learning for unsupervised textual representations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 879–895, Online. Association for Computational Linguistics.

John A Hartigan and Manchek A Wong. 1979. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. series c (applied statistics)*, 28(1):100–108.

Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning distributed representations of sentences from unlabelled data. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1367–1377, San

Diego, California. Association for Computational Linguistics.

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

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 *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8826–8837, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. 2020. Hard negative mixing for contrastive learning. *Advances in Neural Information Processing Systems*, 33:21798–21809.

Taeuk Kim, Kang Min Yoo, and Sang-goo Lee. 2021. Self-guided contrastive learning for BERT sentence representations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2528–2540, Online. Association for Computational Linguistics.

Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020a. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130, Online. Association for Computational Linguistics.

Junnan Li, Pan Zhou, Caiming Xiong, and Steven Hoi. 2020b. Prototypical contrastive learning of unsupervised representations. In *International Conference on Learning Representations*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

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 *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 216–223, Reykjavik, Iceland. European Language Resources Association (ELRA).

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 271–278, Barcelona, Spain.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 115–124, Ann Arbor, Michigan. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Xuan-Hieu Phan, Le-Minh Nguyen, and Susumu Horiguchi. 2008. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *Proceedings of the 17th international conference on World Wide Web*, pages 91–100.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. *Advances in Neural Information Processing Systems*, 30.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. 2021. Whitening sentence representations for better semantics and faster retrieval. *arXiv preprint arXiv:2103.15316*.

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

Hao Wang, Yangguang Li, Zhen Huang, Yong Dou, Lingpeng Kong, and Jing Shao. 2022a. Sncse: Contrastive learning for unsupervised sentence embedding with soft negative samples. *arXiv preprint arXiv:2201.05979*.

Haobo Wang, Ruixuan Xiao, Yixuan Li, Lei Feng, Gang Niu, Gang Chen, and Junbo Zhao. 2021. Pico: Contrastive label disambiguation for partial label learning. In *International Conference on Learning Representations*.

Tongzhou Wang and Phillip Isola. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9929–9939. PMLR.

Wei Wang, Liangzhu Ge, Jingqiao Zhang, and Cheng Yang. 2022b. Improving contrastive learning of sentence embeddings with case-augmented positives and retrieved negatives. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2159–2165.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language resources and evaluation*, 39(2):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 *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3898–3907, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Zhuofeng Wu, Sinong Wang, Jiatao Gu, Madian Khabsa, Fei Sun, and Hao Ma. 2020. Clear: Contrastive learning for sentence representation. *arXiv preprint arXiv:2012.15466*.

Junyuan Xie, Ross Girshick, and Ali Farhadi. 2016. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487. PMLR.

Jiaming Xu, Bo Xu, Peng Wang, Suncong Zheng, Guanhua Tian, and Jun Zhao. 2017. Self-taught convolutional neural networks for short text clustering. *Neural Networks*, 88:22–31.

Hong Xuan, Abby Stylianou, Xiaotong Liu, and Robert Pless. 2020. Hard negative examples are hard, but useful. In *European Conference on Computer Vision*, pages 126–142. Springer.

Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021. ConSERT: A contrastive framework for self-supervised sentence representation transfer. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5065–5075, Online. Association for Computational Linguistics.

Bo Yang, Xiao Fu, Nicholas D Sidiropoulos, and Mingyi Hong. 2017. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *international conference on machine learning*, pages 3861–3870. PMLR.

Jianhua Yin and Jianyong Wang. 2016. A model-based approach for text clustering with outlier detection. In *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, pages 625–636. IEEE.

Jiali Zeng, Yongjing Yin, Yufan Jiang, Shuangzhi Wu, and Yunbo Cao. 2022. Contrastive learning with prompt-derived virtual semantic prototypes for unsupervised sentence embedding. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 7042–7053, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Dejiao Zhang, Shang-Wen Li, Wei Xiao, Henghui Zhu, Ramesh Nallapati, Andrew O. Arnold, and Bing Xiang. 2021a. Pairwise supervised contrastive learning of sentence representations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5786–5798, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Dejiao Zhang, Feng Nan, Xiaokai Wei, Shang-Wen Li, Henghui Zhu, Kathleen McKeown, Ramesh Nallapati, Andrew O. Arnold, and Bing Xiang. 2021b. Supporting clustering with contrastive learning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5419–5430, Online. Association for Computational Linguistics.

Xiang Zhang and Yann LeCun. 2015. Text understanding from scratch. *arXiv preprint arXiv:1502.01710*.

Yanzhao Zhang, Richong Zhang, Samuel Mensah, Xudong Liu, and Yongyi Mao. 2022a. Unsupervised sentence representation via contrastive learning with mixing negatives. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):11730–11738.

Yuhao Zhang, Hongji Zhu, Yongliang Wang, Nan Xu, Xiaobo Li, and Binqiang Zhao. 2022b. A contrastive framework for learning sentence representations from

pairwise and triple-wise perspective in angular space. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4892–4903, Dublin, Ireland. Association for Computational Linguistics.

Kun Zhou, Beichen Zhang, Xin Zhao, and Ji-Rong Wen. 2022. Debiased contrastive learning of unsupervised sentence representations. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6120–6130, Dublin, Ireland. Association for Computational Linguistics.

A Training Details

We do grid search for the hyperparameters and list the searching space below.

- Total batch size [256, 512]
- Learning rate [1e-5, 3e-5, 5e-5]
- Hard negative weight μ [1.0]
- Number of cluster K [96, 128, 256]
- Momentum γ [1e-3, 5e-4, 1e-4]
- Similarity threshold σ [0.2 0.3 0.4 0.5 0.6]
- Weight of $\mathcal{L}_{bml}$ [1e-2, 1e-3, 1e-4, 1e-5]
- Upper Bound of $\mathcal{L}_{bml}$ α [0, 0.05, 0.1, 0.15, 0.2, 0.25]
- Lower Bound of $\mathcal{L}_{bml}$ β [0.3, 0.4, 0.5, 0.6]

Our method has two main improvements on hard negative and false negative, respectively. We apply both improvements to most of the models except one of them. We list the information in detail in Table 7. More hyperparameter experiments are discussed in Appendix E.

Non-Prompt	BERT		RoBERTa
Models	Base	Large	Base
Hard Negative	✓	✓	✓
False Negative	✓	✓	✓
Prompt-based	BERT		RoBERTa
Models	Base	Large	Base
Hard Negative	✓	✓	
False Negative	✓	✓	✓

Table 7: Hyperparameter settings that whether to apply both improvements on the models.

B Transfer Tasks

Following previous works, we also evaluate our models on seven 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). We evaluate with *Non-Prompt* ClusterNS models, and use the default configurations in SentEval Toolkit. Results are showed in Table 8. Most of our models achieve higher performance than SimCSE and the auxiliary MLM task also works for our methods.

C False Negative Details

We show the case study in Table 10. As we mentioned in Section 5, our method is able to cluster sentences with similar topics such as religion and music, demonstrating that clustering captures higher-level semantics. However, intra-cluster sentences do not necessarily carry the same meaning and thus they are not suitable to be used as positives directly.

We also show the variation tendency of the false negative rate in Figure 7, which is equivalent to the sample percentage of clusters having more than two elements (i.e., the intra-cluster members are the false negatives of each other). We observe that the false negative rate maintains a high percentage in the whole training process, which verifies the necessity to specific handling the false negatives.

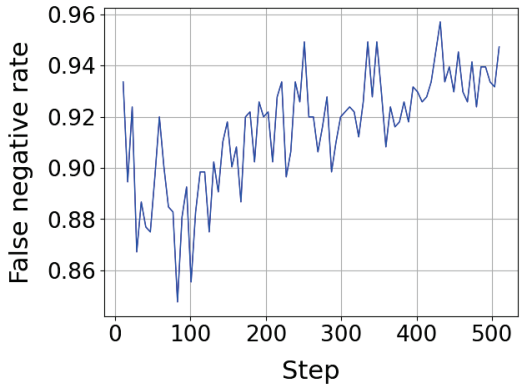


Figure 7: Variation for false negatives rate of *Non-Prompt* ClusterNS-RoBERTa$_{\text{base}}$ in the training process.

D Clustering Evaluation Details

We use adjusted mutual information (AMI) score or clustering accuracy to evaluate clustering performance. AMI score measures the agreement between ground truth labels and clustering results. Two identical label assignments get the AMI score of 1, and two random label assignments are expected to get AMI score of 0. Clustering accuracy measures the clustering agreement with accuracy metric, which need to map clustering results to

Model	MR	CR	SUBJ	MPQA	SST	TREC	MRPC	Avg
GloVe embeddings	77.25	78.30	91.17	87.85	80.18	83.00	72.87	81.52
Avg. BERT embeddings	78.66	86.25	94.37	88.66	84.40	92.80	69.54	84.94
BERT-[CLS] embedding	78.68	84.85	94.21	88.23	84.13	91.40	71.13	84.66
SimCSE-BERT$_{\text{base}}$	81.18	86.46	94.45	88.88	85.50	89.80	74.43	**85.81**
w/ MLM	**82.92**	**87.23**	**95.71**	88.73	86.81	87.01	**78.07**	86.64
ClusterNS-BERT$_{\text{base}}$	82.01	85.46	94.44	**89.09**	86.27	88.80	73.57	85.66
w/ MLM	82.79	86.84	95.29	88.04	**86.88**	**91.80**	76.99	**86.95**
SimCSE-RoBERTa$_{\text{base}}$	81.04	87.74	93.28	86.94	86.60	84.60	73.68	84.84
w/ MLM	83.37	87.76	**95.05**	87.16	**89.02**	90.80	75.13	86.90
ClusterNS-RoBERTa$_{\text{base}}$	81.78	86.65	93.21	**87.85**	87.53	84.00	76.46	**85.35**
w/ MLM	**83.51**	**88.11**	94.56	86.04	88.85	**92.40**	**76.70**	**87.17**

Table 8: Transfer task results of different sentence embedding models. Best results are highlighted in bold.

ground truth labels with Hungary algorithm in advance.

E Supplement Experiments

E.1 Batch Size and Cluster Number

We use large batch sizes and the cluster number K for our models in the main experiments. To show the necessity, we implement the quantitative analysis to compare with small batch sizes and cluster numbers, and show the results in Figure 8 and Figure 9. Both experiments of small batch sizes and cluster numbers perform worse. We attribute the performance degeneration to three factors: 1) Contrastive learning requires large batch sizes in general; 2) Smaller cluster numbers lead to more coarse-grained clusters, weakening the clustering performance; And 3) small batch sizes further restrain the number of clusters.

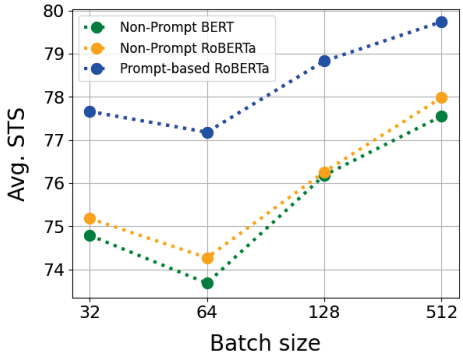


Figure 8: Comparisons of different batch size.

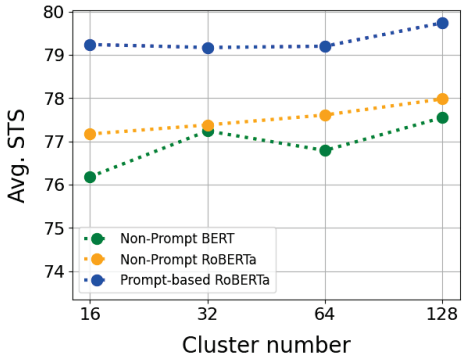


Figure 9: Comparisons of different cluster number.

E.2 Centroids Initialization

We initialize the cluster centroids locally as mentioned in Section 3.3. While some other works adopt global initialization (Li et al., 2020b), they take the embeddings of whole dataset to initialize the centroids. We compare the two strategies by implementing the global initialization version of ClusterNS (named global ClusterNS). We show the test results in Table 9, and the variation of clustering similarity in Figure 10. Overall, global ClusterNS does not improve the performance. We obverse that inter-centroid pairs have extreme high similarity, meaning that clusters do not scatter, and the similarity of hard negative pairs is very low, which means hard negatives are not able to provide strong gradient signal.

Models	Avg. STS
ClusterNS-RoBERTa$_{\text{base}}$	77.98
Global ClusterNS-RoBERTa$_{\text{base}}$	77.81

Table 9: Comparsion of different centroid initialization methods with *Non-Prompt* ClusterNS-RoBERTa$_{\text{base}}$.

Example 1
#1: Jantroon as a word is derived from an Urdu word [UNK] which means Paradise.
#2: While the liturgical atmosphere changes from sorrow to joy at this service, the faithful continue to fast and the Paschal greeting, "Christ is risen!
#3: There is also a Methodist church and several small evangelical churches.
#4: Hindu Temple of Siouxland
#5: Eventually, the original marble gravestones had deteriorated, and the cemetery had become an eyesore.
#6: Reverend Frederick A. Cullen, pastor of Salem Methodist Episcopal Church, Harlem's largest congregation, and his wife, the former Carolyn Belle Mitchell, adopted the 15-year-old Countee Porter, although it may not have been official.
#7: The also include images of saints such as Saint Lawrence or Radegund.
Example 2
#1: Besides Bach, the trio recorded interpretations of compositions by Handel, Scarlatti, Vivaldi, Mozart, Beethoven, Chopin, Satie, Debussy, Ravel, and Schumann.
#2: Guitarist Jaxon has been credited for encouraging a heavier, hardcore punk-influenced musical style.
#3: Thus, in Arabic emphasis is synonymous with a secondary articulation involving retraction of the dorsum or root of the tongue, which has variously been
#4: MP from January, 2001 to date.
#5: The song ranked No.
#6: The tones originate from Brown's acoustic Martin guitar, which is set up through two preamplifiers which are connected to their own power amplifiers.

Table 10: Illustrative examples in clusters resulting from ClusterNS. Sentences with similar topics are grouped into clusters.

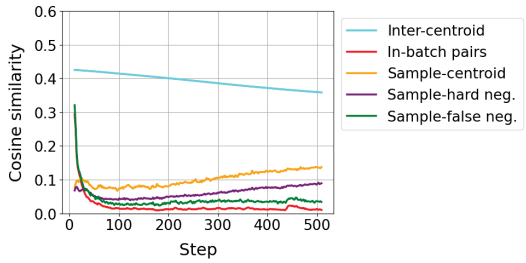


Figure 10: Variation for similarity of sample-nearest centroid pairs (Sample-centroid), sample-hard negative pairs, inter-centroid pairs, intra-cluster member pairs and in-batch negative pairs in global ClusterNS, corresponding to Figure 4.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
After the Conclusion section.

☑ A2. Did you discuss any potential risks of your work?
We discuss them in Limitation.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and Introduction (Section 1).

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

We describe them in Section 4.1 (Evaluation Setup) and Section 4.2 (Implementation Details).

☑ B1. Did you cite the creators of artifacts you used?
We describe them in Section 4.1 (Evaluation Setup) and Section 4.2 (Implementation Details).

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

☒ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
We follow the same processing as previous works, and the datasets and code we used are compatible with the original conditions.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☒ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
All the datasets we used are public and have standard data splits. We follow the same processing as previous works, which also do not mention the relevant statistics.

C ☑ Did you run computational experiments?

Experiments (Section 4).

☒ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
The models (BERT, RoBERTa) we employ in the paper are well-known.

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

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
We discuss the experimental setup in Evaluation Setup (Section 4.1) and Implementation Details (Section 4.2) and discuss the hyperparameter in Training Details (Appendix A) and Hyperparameters Choice

☒ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
We report the main results in a single run and repeat 5 random seeds experiments for our models later. Our standard deviation is about 0.1 0.2 and improvement is also significant statistically.

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
We discuss it in Implementation Details (Section 4.2)

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

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.