HS-GC: Holistic Semantic Embedding and Global Contrast for Effective Text Clustering

Chen Yang, Bin Cao, Jing Fan*
College of Computer Science
Zhejiang University of Technology, Hangzhou, China
{yangchen, bincao, fanjing}@zjut.edu.cn

Abstract

In this paper, we introduce Holistic Semantic Embedding and Global Contrast (HS-GC), an end-to-end approach to learn the instance- and cluster-level representation. Specifically, for instance-level representation learning, we introduce a new loss function that exploits different layers of semantic information in a deep neural network to provide a more holistic semantic text representation. Contrastive learning is applied to these representations to improve the model's ability to represent text instances. Additionally, for cluster-level representation learning we propose two strategies that utilize global update to construct cluster centers from a global view. The extensive experimental evaluation on five text datasets shows that our method outperforms the state-of-the-art model. Particularly on the SearchSnippets dataset, our method leads by 4.4% in normalized mutual information against the latest comparison method. On the StackOverflow and TREC datasets, our method improves the clustering accuracy of 5.9% and 3.2%, respectively.

Keywords: Deep text clustering, Contrastive learning, Unsupervised learning

1. Introduction

Over the past decades, a large number of clustering algorithms have been proposed, such as K-Means (MacQueen, 1965), Spectral Clustering (Ng et al., 2001), Gaussian Mixture Model(Celeux, 1995). These algorithms are effective only when instances are already represented in a latent vectorized space with a good shape (Zhou et al., 2022), while their performance on the high dimensional data, such as texts, is usually limited due to the poor power of feature learning (Steinbach et al., 2004). Hence, learning representation is a crucial challenge to the text clustering (Yaling Tao, 2021).

Traditionally, representation learning methods model a text sample as bag-of-words (BOW) or term-frequency inverse-document-frequency (TF-IDF). However, BoW and TF-IDF often yield very sparse representation vectors, leading to poor performance (Xu et al., 2017). With the aid of the deep learning (LeCun et al., 2015), deep text clustering methods (Xie et al., 2016; Xu et al., 2017; Huang et al., 2020; Hadifar et al., 2019) are proposed and have achieved promising performance because of the strong representation capability of deep neural networks (Bengio et al., 2013). Despite the promising improvements in the clustering performance, it is still inadequate. One reason is that, even with a deep neural network, there is still significant overlap among clusters in the texts (Zhang et al., 2021). Therefore, most of the existing works (Zhang et al., 2021; Yaling Tao, 2021; Li et al., 2021; Huang et al., 2022) focus on solving the overlap problem between clusters, and the widely used method is contrastive learning.

Contrastive learning has achieved remarkable performance in unsupervised representation learning (Chen et al., 2020a; He et al., 2020; Gao et al., 2021). Its basic idea involves the creation of positive and negative pairs for each instance through data augmentations. These pairs are then projected into a representation space to minimize the distance of positive representation pairs and maximize the distance of negative pairs. This advantageous characteristic can be utilized to facilitate clustering by separating the overlapping clusters (Wang, 2020). As a result, some contrastive learning based clustering methods (Zhang et al., 2021; Yaling Tao, 2021; Li et al., 2021; Huang et al., 2022) are proposed, which achieve state-of-the-art results. These methods usually define two types of loss functions to boost clustering performance, i.e., instance-level loss function and cluster-level loss function to learn instance- and cluster-level representation, respectively. To learn instance-level representations, the feature embedding of a single layer, usually the last fully-connected layer, is used to implement the instance-level loss function. Recent studies (Zeiler, 2014; Jawahar et al., 2019) have demonstrated that different layers in a deep neural network can reflect different levels of semantic information. Hence, combining the feature embedding of different layers can yield more holistic semantic information on the text instances. However, the existing methods often overlook the potential advantages of leveraging the diverse information of multiple network layers for improving the clustering

*Corresponding author

LREC-COLING 2024, pages 8349–8359
20-25 May, 2024.

performance.

Moreover, these methods explicitly design a clustering task for cluster-level loss function to focus on the cluster-level semantic information and try to bring together instances from the same semantic cluster. Nevertheless, these methods usually learn the cluster-level representation from the perspective of mini-batch, i.e., local view. Notably, the cluster itself is a global structure of the dataset (Zhang et al., 1996). Therefore, learning cluster-level representation from a local view without considering the entire dataset, i.e., the global view, may affect the performance of clustering.

To solve these problems, in this paper, we propose a novel method, **HS-GC**: Holistic Semantic Embedding and Global Contrast for Effective Text Clustering, to learn more distinctive instance- and cluster-level representation in an end-to-end manner. For instance-level representation, we design a new loss function to facilitate the model to capture more holistic semantic information in text instances. Specifically, text features are modeled by exploiting different layers of feature embedding in a deep neural network to provide a more holistic semantic text representation. Further, contrastive learning is applied to these representations to improve the model's ability to represent text instances.

Additionally, representing the global information of a dataset at the cluster-level poses a significant challenge. To address this challenge, we propose two strategies: (1) $average\ update$ and (2) $momentum\ update$. Both methods leverage the average of the cluster centers over multiple mini-batches to represent the global information. Here, we call the cluster centers in each mini-batch as the local cluster centers, while the average of all local cluster centers across multiple batches is considered as the global cluster centers. As illustrated in Fig. 1(a), when calculating global cluster centers for the $(t+1)th$ mini-batch, the average update averages the local cluster centers of the first t mini-batches with those from the $(t+1)th$ batch by giving each batch the same weight. As for the $momentum\ update$, we use a momentum-based moving average (Goodfellow et al., 2016), which can be approximated as the weighted average value of the past periods, to update the global cluster centers to ensure the globality of the dataset, as shown in Fig. 1(b). Then, based on these two strategies, we propose to perform contrastive learning over representations of cluster centers, where two augmented views of the same cluster centers are positive pairs and other cluster centers are negative pairs. Benefitting from the novel instance- and cluster-level contrastive loss, our method can learn discriminative features and boosts the clustering performance. In summary, our contributions are as follows:

- We propose a novel end-to-end deep clustering method, HS-GC, based on novel instance- and cluster-level contrastive loss.
- To enhance the comprehensiveness of semantic information in the text instance representation, we employ different layers of feature embedding in a deep neural network.
- We propose two strategies to update the cluster centers from global view.
- We conduct extensive experiments on five well-known public text datasets and ablation studies to demonstrate the effectiveness of the proposed method.

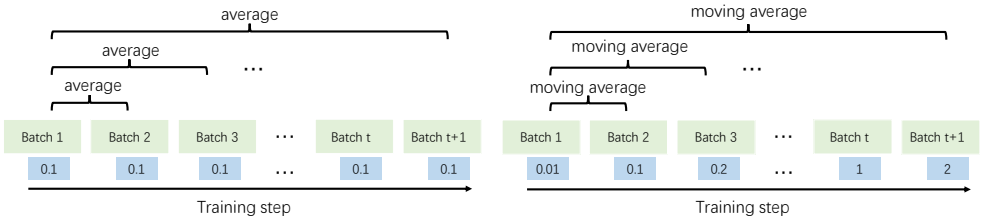
The remainder of this paper is organized as follows. A brief review of the related work of self-supervised representation learning and deep clustering is given in Sec.2. Sec. 3 presents our proposed method. Experimental results are reported and analyzed in Sec. 4. Finally, Sec. 5 concludes the paper.

2. Related Work

This section introduces current self-supervised representation learning methods and reviews the latest deep clustering approaches.

2.1. Self-supervised Representation Learning

Self-supervised representation learning aims to generate semantically meaningful features from samples without requiring human annotations. Previous studies have attempted to capture data features through Auto-Encoder (Vincent et al., 2010), generative model (Kingma, 2013), or heuristic pretext task (Devlin et al., 2019; Li et al., 2020; Yang et al., 2019). Recently, contrastive learning combined with data augmentations strategies has achieved great success. Contrastive learning learns representations of data by contrasting positive pairs of examples against negative pairs. The generation of positive instance features and negative instance features for each sample is critical for constructing the contrastive loss. Different contrastive learning methods vary in their strategy to generate instance features. The memory bank approach (Wu et al., 2018) stores the features of all samples calculated in the previous step. The end-to-end approach (Chen et al., 2020a; Ye et al., 2019) generates instance features using all samples within the current mini-batch. The momentum encoder approach (He et al., 2020) encodes samples on-the-fly by a momentum-updated encoder while maintaining a queue of instance features. Gao et al. (2021) uses standard dropout



(a) Average update (b) Momentum update

Figure 1: Illustration of average update and momentum update on global cluster centers, where the green boxes denote the local cluster centers in each batch, and the blue boxes represent the weights of each batch. The average update gives each batch the same weight to update the global cluster centers. The momentum update uses a batch-by-batch weight increment strategy to update the global cluster centers.

as minimal data augmentation and feeds an identical sample to a model twice with independently sampled dropout masks to generate two distinct instance features as a positive pair. However, these methods fail to use the semantic information of different layers in the deep neural network to provide more holistic semantic representation on instances.

2.2. Deep Clustering

Benefitting from the powerful representation capability of deep neural networks, researchers adopt deep neural network to extract friendly features for clustering and has shown promising performance. Deep Embedded Clustering (DEC) (Xie et al., 2016) trains the autoencoder with the reconstruction loss and optimizes KL-divergence with an auxiliary target distribution to refine cluster centers iteratively. Deep clustering Network (DCN) (Yang et al., 2017) further introduces a k-means loss as the penalty term to reconstruct the clustering loss. Xu et al. (2017) use a convolutional neural network to learn the deep feature representation, then K-Means is deployed to obtain clustering results. Following (Xie et al., 2016; Hadifar et al., 2019), further enhances sample representation using smooth inverse frequency(Arora et al., 2017). Despite the promising improvements in the clustering performance, there is still significant overlap among clusters in the dataset (Zhang et al., 2021). Recently, contrastive learning based on deep clustering has achieved excellent performance. Contrastive Clustering (CC) (Li et al., 2021) jointly performs instance- and cluster-level contrastive learning. However, CC implements cluster-level loss on cluster probabilities, which may result in the loss of semantic information in the learned representation (Huang et al., 2022). Prototype scattering and positive sampling (Propos) (Huang et al., 2022) implements the contrastive loss on the representation of the cluster centers within a mini-batch. However, CC and Propos both implement cluster-level contrastive loss from the perspective of mini-batch, which may mislead final clustering results. Supporting Clustering with Contrastive Learning (SCCL) (Zhang et al., 2021) uses contrast learning for instance-level representation learning and sharpened soft assignment for cluster-level representation learning. Soft assignment distribution describes the similarity between the instance and each cluster centers. Regrettably, the soft assignments are prone to the influence of cluster centers initialization. Moreover, although SCCL initializes the cluster centers from a global perspective, it learns cluster-level representation from a local perspective in training phase without considering the global structure of the dataset.

3. Methodology

In this section, we present our method for deep text clustering. As illustrated in Fig.2, our method is composed of three jointly learned components, namely, a text representation backbone (**TRB**), and an instance-level contrastive head (**ICH**) and a cluster-level global contrastive head (**CGCH**). Briefly speaking, The TRB utilizes data augmentations to generate augmented pairs of text, where data consists of both the two augmented and the original texts, i.e. $\{x^a, x^b, x\}$. The triplet is then processed using a BERT-like pre-trained model $\psi(\cdot)$, to extract features, i.e., $\{h^a, h^b, h\}$, where $\{h^a, h^b\}$ from the two augmented samples are subsequently used for contrastive learning in ICH and h from the original sample is used to generate pseudo-labels via a clustering algorithm denoted as $f(\cdot)$. Then CGCH is utilized for cluster-level contrastive learning on the cluster centers $\{\mu^a, \mu^b\}$ formed by $\{h^a, h^b\}$. Next, we will provide detailed explanations for each of the three components.

3.1. Text Representation Backbone

The architecture of Transformer (Vaswani et al., 2017) is the basis for the development of modern pre-trained language models (PLM) (Devlin et al., 2019; Brown et al., 2020). Given an input

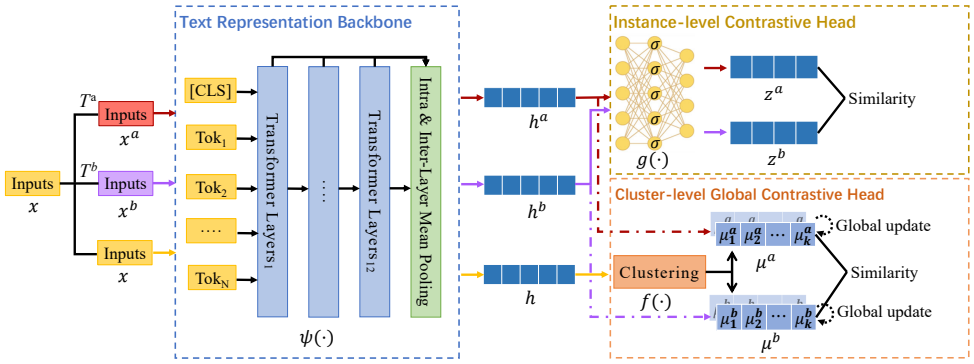


Figure 2: The model architecture of the proposed HS-GC.

sentence $x = \{t_1, ..., t_n\}$, an L-layer Transformer-based PLM yields a seires of hidden embedding representations $\mathcal{E} = \{E_0, E_1, E_2, ..., E_L\}$, where $E_i = [e_i^1, e_i^2, ..., e_i^n](0 < i \leq L)$ are the embedding vectors for each token in x in the i-th Transformer layer. The hidden representation E_L^0 of [CLS] token is used to represent one sentence and is fed into the classification layer for fine-tuning. However, in the unsupervised text clustering settings, the absence of labeled data makes it impossible to fine-tune our model, resulting in the [CLS] token's hidden vector potentially missing out on vital information (Huang et al., 2021; Li et al., 2020). Thus in existing works (Zhang et al., 2021; Huang et al., 2020), mean-pooled E_L, i.e., mean pooling on token embeddings from the last hidden layer, is usually used to synthesize the text representation for further clustering. However, different layers in a BERT-like pre-trained model can reflect different levels of semantic information (Jawahar et al., 2019) and a combination of these layers can provide a more holistic understanding of the text instances . Hence, we propose the following strategies to derive more holistically semantic text representations:

Intra-layer mean pooling: We calculate the average hidden vectors for all tokens within each layer as the representation p_i of the i-th layer by Eq. 1,

$$p_i = \frac{1}{n} \sum_{j=1}^{n} e_i^j \tag{1}$$

Inter-layer mean pooling: We further perform mean pooling on all pooled layer representations $p_1, p_2, ..., p_L$ to obtain the final text representation h by Eq. 2.

$$h = \frac{1}{L} \sum_{i=1}^{L} p_i \tag{2}$$

3.2. Instance-level Contrastive Head

Contrastive learning aims to learn effective representation by maximizing the similarities between positive pairs and minimizing the similarities between negative pairs. Various criteria can be used to define these pairs. For instance, positive pairs can be defined as samples within the same class, while negative pairs include those from different classes. However, since the absence of prior label information in text clustering settings, positive and negative pairs are generated at the instance-level by utilizing data augmentations. Specifically, positive pairs consist of augmented samples from the same instance, while negative pairs are all the other examples in the sampled mini-batch.

Formally, given a mini-batch $\mathcal{B}$ of size N, two types of data augmentations, i.e. T^a, T^b, are respectively performed on each instance x_i and result in $2N$ data samples $\{x_1^a, ..., x_N^a\}$ and $\{x_1^b, ..., x_N^b\}$. For a specific sample x_i^a, we choose its corresponding augmented samples x_i^b to construct positive pairs $\{x_i^a, x_i^b\}$, while treating the other $2N$-2 examples in $\mathcal{B}$ as negative instances regarding this positive pair. Then a BERT-like pre-trained model is used to extract features from the augmented samples according the Eq.1 and 2, i.e. $h_i^a = \psi(x_i^a), h_i^b = \psi(x_i^b)$.

In practice (Chen et al., 2020a,b), instead of directly applying the contrastive learning on the representation h, we first map it to another subspace using a two-layer nonlinear MLP $g(\cdot)$ to obtain $z_i = g(h_i)$ and then minimize the following loss function:

$$\ell_{i^a}^I = -\log \frac{\exp\left(\text{sim}\left(z_i^a, z_i^b\right)/\tau_I\right)}{\sum_{j=1}^{2N} \mathbb{1}_{j \neq i^1} \cdot \exp\left(\text{sim}\left(z_i^a, z_j\right)/\tau_I\right)} \tag{3}$$

where τ_I is the temperature parameter that defines the degree of attraction and repulsion between samples and $\mathbb{1}_{j \neq i^1}$ is an indicator function. The pairwise similarity is measured by cosine distance, i.e.,

$$\text{sim}\left(z_i^a, z_j^b\right) = \frac{\left(z_i^a\right)\left(z_i^b\right)^T}{\|z_i^a\| \left\|z_i^b\right\|} \tag{4}$$

The use of z_i instead of h_i in Eq. (3) is justified by the concern that minimizing the contrastive loss

on h_i may result in the loss of important information. Empirical evidence from prior studies (Chen et al., 2020a,b) has also demonstrated that minimizing z_i leads to better results.

The final instance-level contrastive loss function is defined as the average of the loss for all positive pairs across the mini-batch $\mathcal{B}$:

$$\mathcal{L}_{ins} = \sum_{i=1}^{2N} \ell_i^I / 2N \tag{5}$$

3.3. Cluster-level Global Contrastive Head

A well-performing clustering method is expected to exhibit well-separated clusters. Suppose that our data consists of K semantic categories, where K is assumed to be known, a contrastive loss can naturally be constructed for these K clusters. Specifically, for a given cluster center, the remaining K-1 cluster centers serve as negative examples. Therefore, we propose a cluster-level contrastive loss to encourage maximizing the inter-cluster distance and minimizing the intra-cluster distance.

Formally, assume that we obtain K cluster centers in the embedding space from one augmented view, $\{\mu_1^a, ..., \mu_K^a\}$, and other K cluster centers from another augmented view, $\{\mu_1^b, ..., \mu_K^b\}$. Here, the cluster centers μ^a, μ^b are computed within a mini-batch $\mathcal{B}$, i.e., the local cluster centers, as follows:

$$\mu^a = \frac{\sum_{x^a \in \mathcal{B}} p(k \mid x) h^a)}{\left\| \sum_{x^a \in \mathcal{B}} p(k \mid x) h^a) \right\|_2} \tag{6}$$

$$\mu^b = \frac{\sum_{x^b \in \mathcal{B}} p(k \mid x) h^b)}{\left\| \sum_{x^b \in \mathcal{B}} p(k \mid x) h^b) \right\|_2} \tag{7}$$

where $p(k|x)$ is the pseudo-label obtained by using clustering algorithm $f(\cdot)$ on original text x. Then we propose two global update strategies to update cluster centers from the global view: (1) $average\ update$ (2) $momentum\ update$.

Average update: when calculating the global cluster centers under the t-th training step, the average update averages the local cluster centers of the previous $(t-1)$ mini-batches with those from the t-th batch by giving each batch the same weight:

$$v_t = \frac{1}{t} \sum_{i=1}^{t} \mu_t \tag{8}$$

where μ_t denotes the local cluster centers of samples in the mini-batch under the t training step and v_t denotes the global cluster centers under the t-th training step.

Momentum update: when calculating the global cluster centers under the t-th training step, the momentum update averages the local cluster centers of the previous $(t-1)$ mini-batches with those from the t-th batch by batch-by-batch weight increment strategy. More precisely, given a momentum coefficient $m \in [0, 1)$, after each training step we perform the following update:

$$v_t = m v_{t-1} + (1-m)\mu_t \tag{9}$$

where v_{t-1} denotes the global cluster centers for the $(t-1)$-th training step, μ_t denotes the local cluster centers for the current training step t and v_t denotes the global cluster centers under the t-th training step.

Thus, we can obtain the global cluster centers v^a from one augmented view and v^b from another augmented view through the Eq. 8 or Eq. 9. According to the Eq. 3, cluster-level contrastive loss can be naturally defined as follows:

$$\ell_{i^a}^C = -\log \frac{\exp\left(\mathrm{sim}\left(v_i^a, v_i^b\right)/\tau_C\right)}{\sum_{j=1}^{K} \mathbb{1}_{j \neq i^1} \cdot \exp\left(\mathrm{sim}\left(v_i^a, v_j\right)/\tau_C\right)} \tag{10}$$

where τ_C is the cluster-level temperature parameter and $\mathrm{sim}(\cdot)$ follows Eq. 4. By traversing all K clusters, the cluster-level contrastive loss is finally computed by

$$\mathcal{L}_{clu} = \frac{1}{2K} \sum_{i=1}^{K} \hat{\ell}_i^C \tag{11}$$

Overall loss: In summary, our overall loss is

$$\mathcal{L} = \mathcal{L}_{ins} + \lambda \mathcal{L}_{clu} \tag{12}$$

where λ controls the balance between two loss components. Therefore, there are only two hyperparameters including the momentum coefficient m and the loss weight λ in the loss function.

4. Experiments

In this section, we conduct experiments to verify the effectiveness of the proposed method.

4.1. Datasets

In the experiments, we assess our method on five widely used text datasets for text clustering. Table 1 provides an overview of the main statistics. **AgNews** (Rakib et al., 2020), is a subdataset of AG's corpus (Zhang and LeCun, 2015) of news articles constructed by assembling titles and description fields of articles from the 4 largest classes of AG's Corpus. **SearchSnippets** is selected from the results of web search transaction using predefined phrases of 8 different domains, which contains 12,340 snippets (Phan et al., 2008). **StackOverflow**, a sentence dataset published in Kaggle.com, and following the previous works (Xu et al., 2017; Zhang et al., 2021), we use a subset (Xu

et al., 2017) of it, where 20 classes are involved and each class has 1,000 samples. **Biomedical** (Xu et al., 2017) is a subset of the dataset distributed by BioASQ's official website, where 20,000 paper titles from 20 groups are randomly selected. **GoogleNewsTS** consists of the titles and snippets of 11,109 news articles about 152 events from GoogleNews(Yin and Wang, 2016). **TREC** is an open-domain, fact-based questions dataset, which contains six categories and 6,000 examples (Voorhees et al., 1999).

Dataset	$\|V\|$	Documents		Clusters	
		#D	#L	#C	L/S
AgNews	21k	8,000	23	4	1
SearchSnippets	31k	12,340	18	8	7
StackOverflow	15k	20,000	8	20	1
GooglenewsTS	20k	11,109	28	152	143
TREC	8k	6,000	10	6	2

Table 1: A summary of datasets used for evaluations. $|V|$: the vocabulary size; #D: number of text documents; #L: average number of words in each document; #C: number of clusters; L/S: the ratio of the size of the largest cluster to that of the smallest cluster.

4.2. Evaluation Metrics

We follow the previous work (Hadifar et al., 2019; Rakib et al., 2020; Zhang et al., 2021), using two common clustering performance metrics to evaluate our method, i.e., Accuracy (ACC) (Wu, 2006) and Normalized Mutual Information (NMI) (Chen et al., 2010). Larger ACC and NMI indicate better clustering result.

4.3. Experimental Setup

Our method is implemented by PyTorch 1.7.1 (Paszke et al., 2017) and Transformers library (Wolf et al., 2020). We select $distilbert\text{-}base\text{-}nli\text{-}stsb\text{-}mean\text{-}tokens$ as the backbone for a fair comparison (Zhang et al., 2021), and we set the maximum input length to 32. For the instance-level contrastive loss, we optimize an MLP with one hidden layer of size 768 and output vectors of size 128. We used the Adam optimizer (Kingma, 2015) with constant 5e-6 learning rate and 10e-4 weight decay while setting the learning rate to 5e-4 to optimize the instance-level contrastive head. The instance-level temperature parameter τ_I is fixed to 0.7 in all experiments, and cluster-level temperature parameter $\tau_I = 1.0$ is used for all datasets. We fix the momentum coefficient m to 0.9 in all experiments, and the effect of the momentum coefficient can be seen in Subsec.4.5. Our empirical investigations reveal that setting the loss weight $\lambda = 10$ yields comparatively better yet stable performance across datasets. The pseudo-labels are obtained by K-Means. The texts are augmented by Contextual Augmenter (Zhang et al., 2021; Kobayashi, 2018). The experimental results are averaged over five random runs.

4.4. Clustering Performance Comparison

We evaluate the proposed method on five challenging text benchmarks and compare it with the following eight representative state-of-the-art clustering approaches.

- **BoW & TF-IDF** are evaluated by applying K-means (MacQueen, 1965), where the 1,500 most frequently used words as the features of the documents.
- **STCC** (Xu et al., 2017) involves three stages. First, it pretrains word embeddings for each dataset using Word2Vec (Mikolov et al., 2013) on a large in-domain corpus. Then, it optimizes a convolutional neural network to improve representations for the final clustering stage with K-means.
- **Self-Train** (Hadifar et al., 2019) improves word embeddings pretrained by (Xu et al., 2017) using SIF (Arora et al., 2017). It starts by using layer-wise pretraining to create an autoencoder following (Xie et al., 2016). The autoencoder is then fine-tuned with a clustering objective.
- **HAC-SD** (Rakib et al., 2020) utilizes a pairwise similarity matrix, setting scores below a threshold to zero. The final clustering results are obtained through hierarchical agglomerative clustering on this adjusted matrix.
- **CC** (Li et al., 2021) uses data augmentation to create positive and negative instance pairs for contrastive learning at both instance and cluster levels. These pairs are projected into a feature space, where contrastive learning occurs in row and column spaces for instance and cluster levels, respectively.
- **SCCL** (Zhang et al., 2021) jointly optimize a top-down clustering loss with a bottom-up instance-wise contrastive loss, where cluster loss follows the approach proposed by (Hadifar et al., 2019) and contrastive loss follows the approach proposed by (Chen et al., 2020a).
- **ProPos** (Huang et al., 2022) maximize the distance between prototypical representations, improving the uniformity of representations. Then align one augmented view of instance with the sampled neighbours of another view—assumed to be a truly positive pair in the embedding space to improve the within-cluster compactness.

Datasets	AgNews		SearchSnippets		StackOverflow		GooglenewsTS		TREC	
Metrics	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
BoW	27.6	2.6	24.3	9.3	18.5	14.0	57.5	81.9	24.9	3.3
TF-IDF	34.5	11.9	31.5	19.2	58.4	58.7	68.0	88.9	27.6	4.0
STCC	83.5	56.9	77.0	56.6	51.1	49.0	76.9	80.6	37.4	9.3
Self-Train	63.6	35.5	77.1	56.7	59.8	54.8	59.4	79.6	39.5	11.6
HAC-SD	81.8	54.6	82.7	63.8	64.8	59.5	85.8	88.0	31.2	5.4
CC	81.0	56.6	74.6	61.0	68.7	61.9	85.6	93.0	39.3	14.4
SCCL	86.2	64.9	76.6	58.8	66.5	66.2	83.3	93.2	40.1	14.1
ProPos	84.3	59.3	74.3	55.2	64.5	55.3	73.9	90.4	37.8	11.5
HS-GC$_a$	83.1	58.6	74.8	58.2	74.3	66.3	85.5	91.2	37.7	11.9
HS-GC$_m$	**87.7**	**66.9**	**85.0**	**68.2**	**74.6**	**67.1**	**88.1**	**94.3**	**43.3**	**15.7**

Table 2: The clustering performance on five challenging text benchmarks. The best results are shown in boldface.

Table 2 shows the results of our proposed method on five text benchmark datasets compared to the state-of-the-art baselines, where HS-GC$_a$ and HS-GC$_m$ use average update and momentum update to update cluster centers, respectively. As is evident in this table, our proposed method outperforms all other baselines in all datasets. Quantitatively, our method improves the ACC by $1.5\%, 2.3\%, 5.9\%, 2.3\%, 3.2\%$, the NMI by $2.0\%, 4.4\%, 0.9\%, 1.1\%, 1.3\%$ respectively on AgNews, SearchSnippets, StackOverflow, GoogleNewsTS, and TREC. The main difference between our framework and other baselines is that we exploit different levels of semantic information in a deep neural network and global information of dataset enhance the learned representation for clustering. These are why our method's performance is superior to the baselines. Moreover, the performance of HS-GC$_m$ is better than HS-GC$_a$. One reason is that due to the lack of consistency in model parameters under different training steps, the representations of samples in each mini-batch are also inconsistent and the representations of the old mini-batches usually are worse than the representations of the new mini-batches. Hence, giving the local cluster centers of each mini-batch the same weight may leads to biased clustering results. In contrast, HS-GC$_m$ achieves better results by utilizing the batch-by-batch weight increment strategy to update cluster centers to alleviate the problem of the poor representation of the old mini-batches. Moreover, we conduct a statistic significance test based on the experimental results reported in Table 2. We select the Friedman test (Friedman, 1937) that can be used to investigate the difference among the various methods on multiple datasets. The null-hypothesis indicates that all the algorithms are equivalent. Notably, the Friedman test has a p-value of 2e-11 that smaller than 0.0001. That is to say the compared methods are significantly different.

4.5. Ablation Study

In this section, we conduct several ablation studies, including the effect of the proposed loss, the effect of different momentum coefficient on updating cluster centers and the effect of different pooling strategies, to demonstrate the effectiveness of different components in our method.

Effect of the proposed loss To verify the effectiveness of the instance-level contrastive loss, i.e. ICH, and cluster-level momentum contrastive loss, i.e. CGCH, we conduct a set of ablation experiments on five text datasets. Clustering results are obtained by K-Means. The obtained results are presented in Table 3. Remarkably, CGCH exhibits comparable performance on Agnews, StackOverflow and GoogleNewsTS, while ICH perform better on SearchSnippets and TREC. These findings suggest that the combined effects of the two losses are evident to some extent.

Effect of different momentum coefficient m This subsection investigates the effect of different momentum coefficient m on clustering in terms of ACC and ARI and reports the results in Figure 3. To analyze the impact of momentum coefficient m, we conduct an experiment by varying m in $[0, 0.9, 0.99, 0.999, 0.9999]$. When m=0, the global cluster centers are solely constructed from the current mini-batch. For $m > 0$, global cluster centers can be approximately constructed as the weighted average of the local cluster centers in the previous $1/(1-m)$ mini-batches, with the weights increasing batch by batch. As shown in Fig.3, the best performance is achieved when m = 0.99, and either a larger or smaller margin degrades the performance. The results reveal that m values that are too small are insufficient to encompass the entire dataset. In contrast, too large values reduce the influence of the initial few mini-batches since the weight increases batch by batch.

Effect of different pooling strategy This subsection investigates the efficacy of different pooling

Datasets	Agnews		SearchSnippets		StackOverflow		GooglenewsTS		TREC	
Metrics	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
w/o ICH	84.9	61.3	64.8	49.5	70.3	63.7	85.7	88.8	36.1	12.9
w/o CGCH	82.9	61.2	82.7	65.5	54.1	47.9	78.8	91.1	40.5	13.9
HS-GC$_m$	**87.7**	**66.9**	**85.0**	**68.2**	**74.6**	**67.1**	**88.1**	**94.3**	**43.3**	**15.7**

Table 3: Effect of the proposed loss on five text benchmarks. The best results are shown in boldface.

Datasets	Agnews		SearchSnippets		StackOverflow		GooglenewsTS		TREC	
Metrics	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
$last\text{-}avg$	81.7	58.2	71.2	63.0	67.0	59.8	83.8	92.3	36.3	12.2
$first\text{-}last\text{-}avg$	87.5	65.5	73.6	65.1	72.3	65.0	84.5	92.8	37.4	12.0
$all\text{-}avg$	**87.7**	**66.9**	**85.0**	**68.2**	**74.6**	**67.1**	**88.1**	**94.3**	**43.3**	**15.7**

Table 4: The performance of different pooling strategies on five challenging text benchmarks. The best results are shown in boldface.

Figure 3: The performance of different momentum coefficient on five challenging text benchmarks.

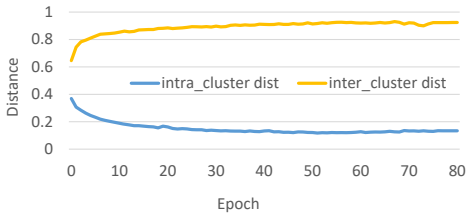
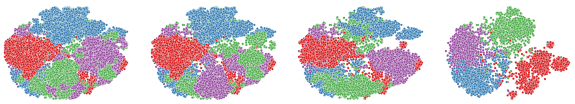


Figure 4: Intra- and inter-cluster distance across the training process on AgNews.



(a) 0 epoch (b) 20 epoch (c) 40 epoch (d) 80 epoch

Figure 5: The evolution visualization of instance features and clustering results across the training process on AgNews. The colors indicate the cluster label obtained from clustering results.

strategies on clustering and reports the results in Table 4. Two commonly used methods, $last\text{-}avg$ and $first\text{-}last\text{-}avg$, are compared, where the former calculates the average of all output vectors in the final layer (Li et al., 2020; Reimers, 2019), and the latter computes the average of all output vectors in the first and final layers (Li et al., 2020; Huang et al., 2021). A novel approach, $all\text{-}avg$, is also proposed, which involves a combination of intra-layer mean pooling and inter-layer mean pooling strategies. Table 4 reveals that the $all\text{-}avg$ approach outperforms the other two methods, indicating that leveraging features from different layers in a deep neural network jointly provides a more comprehensive semantic representation of the data instances and enhances the clustering performance.

4.6. Qualitative Study

We conduct two experiments to analyze the intra- and inter-cluster distance across the training process and the evolution of the learned instance representation and clustering results on AgNews.

Analysis on intra- and inter-cluster distance In order to elucidate the underlying principles of our approach, we present a visualization of the changes of both the intra- and inter-cluster distance w.r.t. the training epoch. An optimal clustering outcome is characterized by low intra-cluster distance and high inter-cluster distance. For a given cluster, the intra-cluster distance is the average distance between the cluster center and all samples belonging to that cluster. The inter-cluster distance is the distance to its nearest neighbouring cluster. In Fig. 4, we provide the mean values of each distance type obtained by averaging across all clusters. The results show that the inter-cluster distance grows while the intra-cluster distance decreases as the training progresses. This observation confirms the ability of our method to compactly group samples within each cluster and effectively separate different clusters from one another.

Evolution of instance feature and clustering results By jointly optimizing the instance- and cluster-level loss function, the model should learn

discriminative representations and desirable clustering results simultaneously. To see how our model converges to the goal, we perform t-SNE (der Maaten, 2008) in the instance embedding space at four different timestamps during the training process. The visualization results, presented in Fig. 5 illustrate the clustering outcomes using different colours to indicate predicted labels. The result shows that the feature representations are initially randomly distributed, and most instances are grouped into a few clusters. However, as the training process continues, clustering results become more reasonable, and the feature representations become more dispersed, forming more distinct clusters.

5. Conclusion

In conclusion, our study demonstrates the effectiveness of the proposed Holistic Semantic Embedding and Global Contrast (HS-GC) approach for deep text clustering. By leveraging the semantic information embedded in different layers of a deep neural network, we have developed a novel loss function that enables a more holistic representation of text instances. This holistic semantic embedding not only enhances the representation power of individual text instances but also improves the overall clustering accuracy. Furthermore, our approach addresses the limitations of existing contrastive learning-based methods by introducing two strategies that consider the global information of the dataset for cluster-level representation learning. These strategies contribute to constructing cluster centers from a global perspective, leading to improved clustering outcomes. The extensive experimental evaluation conducted on five text datasets validates the superiority of our method over state-of-the-art methods. Besides that, we perform ablation studies to demonstrate the effectiveness of proposed method. In the future, we plan to extend our method to semi-supervised clustering.

6. Acknowledgement

This research was partially supported by STI 2030-Major Projects 2021ZD0200400, National Natural Science Foundation of China (62276233 and 62072405) and Key Research Project of Zhejiang Province (2023C01048).

7. References

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *ICLR*. OpenReview.net.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *TPAMI*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *NIPS*, 33:1877–1901.

Celeux. 1995. Gaussian parsimonious clustering models. *Pattern recognition*.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020a. A simple framework for contrastive learning of visual representations. In *ICML*, Proceedings of Machine Learning Research.

Wen-Yen Chen, Yangqiu Song, Hongjie Bai, Chih-Jen Lin, and Edward Y Chang. 2010. Parallel spectral clustering in distributed systems. *TPAMI*.

Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. 2020b. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*.

Van der Maaten. 2008. Visualizing data using t-sne. *Journal of machine learning research*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Milton Friedman. 1937. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the american statistical association*, 32(200):675–701.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *EMNLP*.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning*. MIT press.

Amir Hadifar, Lucas Sterckx, Thomas Demeester, and Chris Develder. 2019. A self-training approach for short text clustering. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 194–199.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *CVPR*.

Junjie Huang, Duyu Tang, Wanjun Zhong, Shuai Lu, Linjun Shou, Ming Gong, Daxin Jiang, and Nan Duan. 2021. Whiteningbert: An easy unsupervised sentence embedding approach. In *EMNLP*.

Shaohan Huang, Furu Wei, Lei Cui, Xingxing Zhang, and Ming Zhou. 2020. Unsupervised fine-tuning for text clustering. In *Proceedings of the 28th international conference on computational linguistics*, pages 5530–5534.

Zhizhong Huang, Jie Chen, Junping Zhang, and Hongming Shan. 2022. Learning representation for clustering via prototype scattering and positive sampling. *TPAMI*.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does bert learn about the structure of language? In *ACL*.

Kingma. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kingma. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. *arXiv preprint arXiv:1805.06201*.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature*.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. In *EMNLP*.

Yunfan Li, Peng Hu, Zitao Liu, Dezhong Peng, Joey Tianyi Zhou, and Xi Peng. 2021. Contrastive clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

J MacQueen. 1965. Some methods for classification and analysis of multivariate observations. In *Proc. 5th Berkeley Symposium on Math., Stat., and Prob*, page 281.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *ICLR*.

Andrew Ng, Michael Jordan, and Yair Weiss. 2001. On spectral clustering: Analysis and an algorithm. *NIPS*, 14.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch.

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.

Md Rashadul Hasan Rakib, Norbert Zeh, Magdalena Jankowska, and Evangelos Milios. 2020. Enhancement of short text clustering by iterative classification. In *Natural Language Processing and Information Systems: 25th International Conference on Applications of Natural Language to Information Systems, NLDB 2020, Saarbrücken, Germany, June 24–26, 2020, Proceedings 25*, pages 105–117. Springer.

Nils Reimers. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP*.

Michael Steinbach, Levent Ertöz, and Vipin Kumar. 2004. The challenges of clustering high dimensional data. *New directions in statistical physics: econophysics, bioinformatics, and pattern recognition*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *NIPS*, 30.

Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*.

Ellen M Voorhees, Dawn M Tice, et al. 1999. The trec-8 question answering track evaluation. In *TREC*.

Wang. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *ICML*. PMLR.

Thomas Wolf, Lysandre Debut, Victor Sanh, and Julien Chaumond etal. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Online. Association for Computational Linguistics.

Wu. 2006. A local learning approach for clustering. In *NIPS*.

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. 2018. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742.

Junyuan Xie, Ross Girshick, and Ali Farhadi. 2016. Unsupervised deep embedding for clustering analysis. In *ICML*. 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*.

Kouta Nakata Yaling Tao, Kentaro Takagi. 2021. Clustering-friendly representation learning via instance discrimination and feature decorrelation. *Proceedings of ICLR 2021*.

Bo Yang, Xiao Fu, Nicholas D Sidiropoulos, and Mingyi Hong. 2017. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *ICML*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *NIPS*, 32.

Mang Ye, Xu Zhang, Pong C Yuen, and Shih-Fu Chang. 2019. Unsupervised embedding learning via invariant and spreading instance feature. In *CVPR*.

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)*. IEEE.

Zeiler. 2014. Visualizing and understanding convolutional networks. In *ECCV*. Springer.

Dejiao Zhang, Feng Nan, Xiaokai Wei, Shang-Wen Li, Henghui Zhu, Kathleen R. McKeown, Ramesh Nallapati, Andrew O. Arnold, and Bing Xiang. 2021. Supporting clustering with contrastive learning. In *NAACL*.

Tian Zhang, Raghu Ramakrishnan, and Miron Livny. 1996. Birch: an efficient data clustering method for very large databases. *ACM sigmod record*, 25(2):103–114.

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

Sheng Zhou, Hongjia Xu, Zhuonan Zheng, Jiawei Chen, Jiajun Bu, Jia Wu, Xin Wang, Wenwu Zhu, Martin Ester, et al. 2022. A comprehensive survey on deep clustering: Taxonomy, challenges, and future directions. *arXiv preprint arXiv:2206.07579*.