IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 29 June 2022, accepted 14 July 2022, date of publication 19 July 2022, date of current version 25 July 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3192442

RESEARCH ARTICLE

Clustering of Short Texts Based on Dynamic Adjustment for Contrastive Learning

RUIHUI LI AND HONGBIN WANG
Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650500, China
Key Laboratory of Artificial Intelligence, Kunming University of Science and Technology, Kunming 650500, China

Corresponding author: Hongbin Wang (whbin2007@126.com)

This work was supported by the National Natural Science Foundation of China under Grant 61966020.

ABSTRACT Faced with the large amount of unlabeled short text data appearing on the Internet, it is necessary to categorize them using clustering that can divide text into several clusters based on similarity degree of text semantics. Recently, combining clustering with contrastive learning has been the focus of clustering research. Due to the excellent representation learning ability of contrastive learning, clustering achieves better results on short texts that are high-dimensional and sparse. However, contrastive learning pays more attention to general feature representation at the instance-level and ignores the semantic-level correlation of data belonging to same cluster in clustering. The inconsistent training objectives of contrastive learning and clustering lead to lower confidence of clustering results and sparse cluster space. To improve this problem, we propose a clustering method based on Dynamic Adjustment for Contrastive Learning (DACL). The method smoothly transitions loss weight of model from contrastive learning to clustering during training and filters negative samples in contrastive learning by the pseudo-labels generated by clustering. To demonstrate the effectiveness of the method, DACL is compared with eight short text clustering models on eight datasets. The results show that we achieve considerable performance improvements on most datasets compared to state-of-the-art short text clustering methods. In addition, The effectiveness of loss smooth transition and negative filtering is proved by ablation experiments.

INDEX TERMS Contrastive learning, deep clustering, dynamic adjustment, short text clustering.

I. INTRODUCTION

With development of self-media and popularity of social media, short texts have become a common form of content on the Internet. Categorizing these short texts is an important step for other data mining tasks, such as public opinion analysis and event detection [1]. Due to large number of texts and high cost of labeling, it is necessary to categorize them using unsupervised methods. Clustering is one of the important unsupervised methods, that can divide text into several clusters based on similarity degree of text semantics. Traditional clustering methods, such as K-Means [2] and Gaussian Mixture Model (GMM) [3], usually rely on distance measures of samples in initial dimension space to calculate similarity. Because of insufficient feature representation capability, distance measure often does not represent semantics similarity between samples well when facing short text data with high sparsity, high noise and high dimensionality [4]. Recently, some works combine clustering with deep representation learning to achieve better clustering results by embedding original data into low-dimensional space [5].

Contrastive learning has recently achieved remarkable results in deep representation learning [6]. In the unsupervised context, contrastive learning usually considers each instance as a category represented by a feature vector. Positive pairs are constructed by data augmentation, and other instances are treated as negative instances. Feature representation of samples is learned by maximizing similarity of positive pairs from same sample and minimizing similarity of negative instances from other samples. Unlike other self-supervised methods that perform representation learning by reconstructing or generating original data, such as Auto-Encoder (AE) [7] and Generative Adversarial Networks (GAN) [8], contrast learning performs representation

The associate editor coordinating the review of this manuscript and approving it for publication was Arianna D'Ulizia.

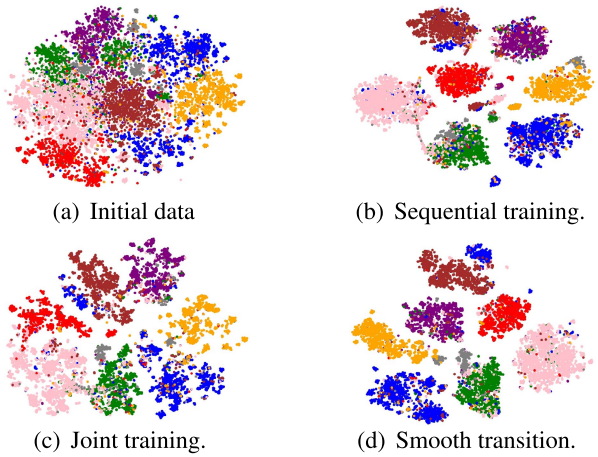

FIGURE 1. Visualize TSNE in embedded spatial representation of SeachSnippets dataset. Each point represents each sample, and each color represents each true semantic cluster.

learning based on the computation of distance measures in the feature space like clustering, so the feature representations obtained by contrast learning are more friendly to clustering. Some works combine clustering with contrastive learning to obtain better clustering results [9]. However, as shown in Figure 1, contrastive learning makes sample distribution scattered and clustering makes sample close to the cluster center. Clustering tasks require samples in the same cluster to be as similar as possible, that is, samples are as small as possible from the semantic center they belong to and as large as possible from other semantic centers. The purpose of comparative learning is to learn the general characteristics of samples. Each instance is regarded as a category, and all other instances are regarded as negative instances. As a result, samples belonging to the same semantic cluster are regarded as negative instances of each other, and samples cannot learn the overall semantic information of clusters and the semantic information differences between clusters.

The inconsistent training objectives of contrast learning and clustering, resulting in a bias between the final training results and the requirements of clustering tasks. The samples treat other samples in the same cluster as negative examples leading to sparse space within the cluster, which is not conducive to the learning of cluster semantic information. In order to solve the problem of existing comparative clustering methods, we propose a clustering method of short texts based on Dynamic Adjustment for Contrastive Learning (DACL). To solve the problem of inconsistency of training objectives, DACL is designed a new adjustment function to dynamically adjust the weights of contrast loss and clustering loss in the total loss function. By adjusting the function, the focus of model training smoothly transitions from contrast learning to cluster, which enables samples to learn semantic information of clusters and improves the confidence of sample cluster assignment probability. To avoid samples from the same cluster becoming negative instances, we introduce the semantic information of clusters into contrastive learning. DACL assigns clusters of samples as pseudo labels and excludes instances with the same label from negative instances. In addition, due to possible errors in the assignment of pseudo-labels, resulting in cumulative errors in subsequent training, only samples with high confidence are assigned pseudo-labels, and the remaining samples are still considered a category for each instance. To summarize, the major contributions of our work are as follows:

- We propose a method to dynamically adjust the loss weights to mitigate the inconsistency of training objectives between contrast learning and clustering. During the training process, the weight of contrastive loss and clustering loss in total loss is adjusted by the dynamic adjustment function to achieve a smooth transition from representation learning to cluster feature learning.
- We introduce the cluster assignment information of samples into the contrast learning to filter negative instances by generating pseudo-labels with high confidence in the cluster assignment probability. By this method, the problem of samples being selected as negative instances from the same cluster is avoided, and the quality of negative instances is effectively improved, making the data representation of contrast learning more friendly to clustering.
- We deeply analyzed and demonstrated the effectiveness of DACL through comparative experiments and ablation experiments. Compared with state-of-the-art short text clustering models, DACL achieves better performance on most of the datasets.

The rest of this paper is organized as follows. Section II reviews the related research in the field of deep clustering and contrastive learning. Section III presents the architecture of the proposed model. Section IV analyzes and discusses the experiments performed. Finally, the conclusion and future works are drawn in Section V.

II. RELATED WORK

In this section, we give a brief introduction to recent developments on two related topics, deep clustering and contrastive learning.

A. DEEP CLUSTERING

Although traditional clustering methods are widely used, the clustering results on large and complex datasets are not satisfactory due to the lack of representation learning capability. Benefiting from the powerful representation capability of deep neural networks, deep clustering has shown good performance on complex datasets. Yang *et al.* [10] proposed deep clustering network (DCN) combines AE with K-Means for deep clustering learning. Because K-Means is not differentiable, DCN adopts strategy of alternate optimization of network parameters and cluster centers. Xie *et al.* [11] proposed Deep Embedded Clustering (DEC), a method that simultaneously learns feature representations and cluster assignments using deep neural networks. Both DCN and DEC use autoencoders as an auxiliary task for representation learning in clustering, while the features obtained from

autoencoders are based on reconstructing the original data and are not relevant to clustering based on distance metrics. To further improve the clustering performance, some works focus on the pairwise distance relationships of samples in the feature space and train models with them [12], [13]. These models obtain similar samples by K-Nearest Neighbor (KNN) and learn the representation by minimizing the distance to similar samples and maximizing the distance to other samples. Recently, contrast learning based on feature space distance measures has achieved excellent performance in representation learning. Zhang *et al.* [14] proposed Supporting Clustering with Contrastive Learning (SCCL) uses contrast learning for representation learning and sharpened confidence distributions for clustering learning. Clustering is performed by jointly optimizing contrast loss and clustering loss. Xu *et al.* [15] proposed Invariant Information Clustering (IIC) directly maximizing the mutual information between the original samples and augmentation samples based on information bottleneck theory. IIC can effectively filter redundant information and capture invariant information between samples.

Since clustering has no representation learning capability, deep clustering mostly obtains feature representations of samples through self-supervised learning of auxiliary tasks, and features obtained by auxiliary tasks are expected to be friendly to clustering. All the above models train the model by combining the auxiliary and clustering tasks by iterative, sequential or joint training methods. However, the auxiliary task focuses more on representation learning of samples themselves, and clustering focuses more on whole features of clusters formed by samples and the differences between clusters. Due to the inconsistent objectives of the auxiliary task and clustering, iterative or sequential learning corrupts the feature representation of the samples learned by the auxiliary task, while joint learning yields clustering results with low semantic confidence. To moderate the target inconsistency, DACL uses a joint training method of clustering and auxiliary tasks, and a new adjustment function is designed to dynamically adjust the weights of auxiliary task loss and clustering loss to the total loss during training. The focus of training is gradually and smoothly transitioned from representation learning to clustering as the training progresses, which avoids the destruction of the feature representation of the samples and gradually reduces the influence of the auxiliary task during training to improve the confidence of the clustering results.

B. CONTRASTIVE LEARNING

As a promising unsupervised learning paradigm, contrastive learning has recently achieved state-of-the-art performance in representation learning. The basic idea of contrastive learning is to construct positive pairs of samples by data augmentation or labeling, and map them to feature space. Representation learning is learned by maximizing the similarity between positive pairs and minimizing the similarity with other positive pairs. The construction of positive pairs and negative instances has a crucial impact on the performance of comparison learning. Chen *et al.* [16] proposed a Simple Framework for Contrastive Learning of Visual Representations (SimCLR) generates positive example pairs using data augmentation twice and treats other samples in the same batch as negative instances. They also demonstrated experimentally that the number of negative instances has critical impact on performance. To further increase the number of negative examples, Misra *et al.* [17] proposed Pretext-Invariant Representation Learning (PIRL) adds a memory storage module to store feature representation of previous sample, which is used as negative instances when calculating objective loss of the current sample. He *et al.* [18] proposed Momentum Contrast (MoCo) uses a queue to store negative instances and designs two encoders, in which the negative instances encoder is momentum updated by parameter of positive pair encoder. Cai *et al.* [19] sorts the negative instances according to the distance of samples in the feature space, and finds that 95% of the simple negative instances are indispensable, and 0.1% of the most difficult negative instances are unnecessary and even reduce the effect. Li *et al.* [20] proposed Contrastive Clustering(CC) introduced the idea of comparative learning into the cluster. Based on the idea of label as representation, CC performs contrastive learning at the instance level and cluster level in the row space and column space of the feature matrix.

In unsupervised comparative learning, positive pairs are usually acquired through data augmentation, so the selection of negative instances becomes particularly important. The above methods generate positive pairs through data augmentation and consider all other samples as negative instances. All the above models treat all other samples as negative instances, ignoring the semantic relationships between samples. When comparative learning is combined with clustering tasks, samples in the same cluster are inevitably considered negative instances, which reduces the clustering results. To avoid this situation. DACL introduces cluster assignment probability information from clustering into comparative learning. The model assigns false labels to high confidence data based on cluster assignment probability during training and excludes samples with the same labels from negative instances. This method avoids the distance being pulled far in the feature space because of samples belonging to same cluster are selected each other as negative instances, which improves the confidence of cluster allocation.

III. PROPOSED METHOD

In view of the problems in above work, a novel model, DACL proposed for short text cluster, which unifies contrastive learning and clustering into a training framework for end-to-end joint training. As shown in Figure 2, this model consists of three parts, encoder $f(\cdot)$, instance-level contrastive head $g_I(\cdot)$ and semantic-level clustering header $g_C(\cdot)$. Encoder is responsible for embedding the samples into the low dimensional feature space to extract features, the cluster head is responsible for calculating the distribution probability of the

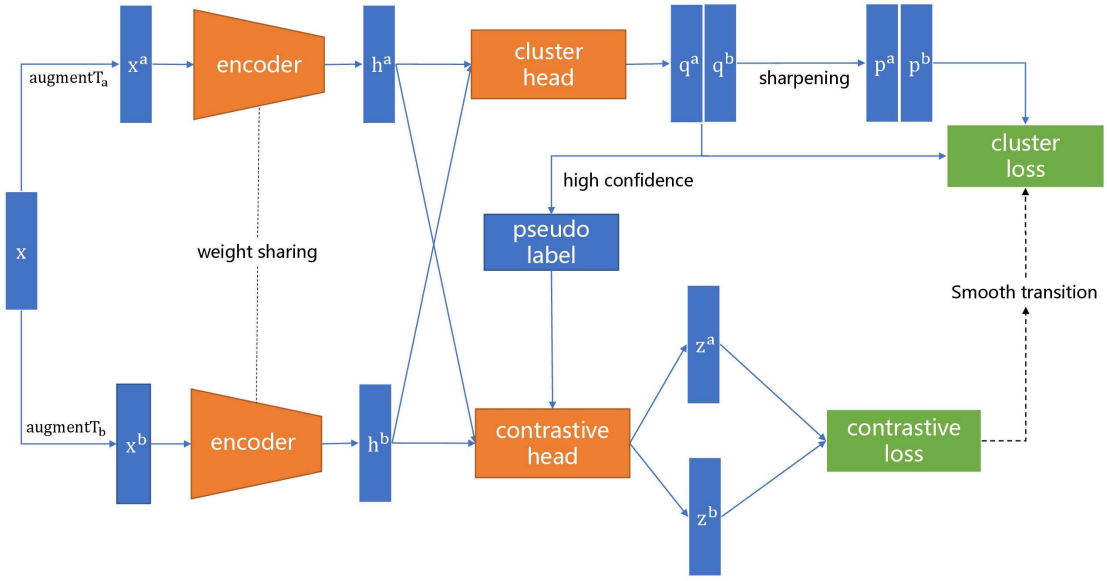

FIGURE 2. Model framework.

samples to each cluster, and the contrastive head is responsible for embedding the samples into the contrastive space. The cluster distribution probability calculated by cluster head can be used to screen the negative instances in the contrastive head. Clustering objective loss and contrastive target loss are used to train the model. Clustering target loss improves the confidence of clustering results by sharpening the cluster distribution probability of samples, and contrastive objective loss optimizes feature representation of samples by maximizing the distance of positive pairs and minimizing the distance of negative instances. In the training process, clustering loss and contrast loss are dynamically added to form the total loss, and the total loss weight gradually transitions from contrast loss to clustering loss. The following subsections describe three components of the model in detail, and finally the proposed objective loss function is described.

A. ENCODER

Inspired by recent contrastive learning work [6], this paper uses data augmentation to generate positive pairs. In view of feature of short length and sparse features of short text, context augmentation [21] is used to augment data. That is, words in the input text are randomly masked and predicted by pre-training language model, and replace original words with the predicted words. Formally, randomly select batch data $\mathcal{B} = \{\boldsymbol{x}_i\}_{i=1}^{M}$ in dataset X, where M is batch size. we use two different pre-training language models T_a and T_b to generate a pair of augmentation data $\boldsymbol{x}_i^a = T_a(\boldsymbol{x}_i)$ and $\boldsymbol{x}_i^b = T_b(\boldsymbol{x}_i)$ for each example in $\mathcal{B}$. Then, deep neural network $f(\cdot)$ with shared parameters is used as encoder to extract features representing $\boldsymbol{h}_i^a = f(\boldsymbol{x}_i^a)$ and $\boldsymbol{h}_i^b = f(\boldsymbol{x}_i^b)$, and finally get augmented batch data $\tilde{B} = \{\boldsymbol{h}_i^a, \boldsymbol{h}_i^b\}_{i=1}^{M}$. As for encoder, theoretically, our method can use any deep representation learning network. In this paper, we choose the same settings as SCCL.

B. SEMANTIC-LEVEL CLUSTER HEAD

The objective of the clustering head is to divide the samples into K disjoint clusters, so that semantics in same cluster are as similar as possible. The semantic center of each cluster is initialized by K-Means to be the centroid $\boldsymbol{\mu}_k$, $k \in \{1, 2, \ldots, K\}$ of cluster in feature space. Thus we evaluate similarity between $\boldsymbol{x}_i$ and $\boldsymbol{\mu}_k$ by calculating Student's t-distribution [22] of $\boldsymbol{x}_i$ in feature space as follows:

$$\boldsymbol{q}_{ik} = \frac{(1 + ||\boldsymbol{h}_i - \boldsymbol{\mu}_k||_2^2/\alpha)^{-\frac{\alpha+1}{2}}}{\sum_{k'=1}^{K}(1 + ||\boldsymbol{h}_i - \boldsymbol{\mu}_{k'}||_2^2/\alpha)^{-\frac{\alpha+1}{2}}} \tag{1}$$

where, α is freedom degree in Student's t-distribution. $\boldsymbol{q}_{ik}$ can be seen as probability of sample x_i is assigned to cluster k, and distribution of soft assignment probability of sample is obtained as follows:

$$\boldsymbol{q}_i = g_C(\boldsymbol{h}_i) = [q_{ik}], \quad k \in \{1, 2, \ldots, K\} \tag{2}$$

After obtaining the cluster soft assignment distribution q_i of sample, our objective is to optimize sample representation by learning from high confidence assignments. So set assistant objective function can be expressed as:

$$p_{ik} = \frac{q_{ik}^2 / \sum_{i=1}^{M} q_{ik}}{\sum_{k'=1}^{K} q_{ik}^2 / q_{ik'}} \tag{3}$$

This objective distribution first sharpens soft assignment probability by raising it to the second power, and then normalizes it by associated cluster frequency [11]. In doing so, we encourage learning from high confidence cluster assignments while countering biases caused by unbalanced clustering.

By Eq.(1) and (3), we obtain respective cluster distribution and assistant distribution $\boldsymbol{q}^a$, $\boldsymbol{q}^b$, $\boldsymbol{p}^a$, $\boldsymbol{p}^b$ from two sets of augmented sample $\boldsymbol{h}^a$ and $\boldsymbol{h}^b$ in B, and then to cross-optimize

the KL divergence between them can be written as:

$$\mathcal{L}_C = \frac{1}{2}(KL[\boldsymbol{p}^b||\boldsymbol{q}^a] + KL[\boldsymbol{p}^a||\boldsymbol{q}^b]) = \frac{1}{2M}\sum_{i=1}^{M}\sum_{k=1}^{K}(p_{ik}^b log\frac{p_{ik}^b}{q_{ik}^a} + p_{ik}^a log\frac{p_{ik}^a}{q_{ik}^b}) \quad (4)$$

The cluster distribution of samples can be pushed towards the objective distribution of its corresponding positive pairs by minimizing the clustering objective loss. In this way, data assignments with higher confidence can be learned, and at the same time, the distribution of positive pairs can be close to each other, thereby improving the clustering quality.

C. INSTANCE-LEVEL CONTRASTIVE HEAD

The objective of contrastive learning is to maximize the similarity of positive pairs while minimizing the similarity of negative instances. Since there are no ground truth labels in the clustering task, we choose data augmentation to construct instance-level data pairs [23]. To avoid negative samples from same cluster, we assign pseudo labels to samples based on cluster soft assignment probability distribution. Samples with same pseudo labels are considered false negative instances and excluded from the negative instances.

Formally, any sample h_i^a in $\tilde{B}$ and corresponding augmented sample h_i^b are selected to become positive pair $\{h_i^a, h_i^b\}$, and negative pair are instances filtered from other data in $\tilde{B}$ through pseudo labels. We assign pseudo labels to data assignments by the cluster soft assignment probability distribution q_i^a and q_i^b obtained by Eq.(2) as follows:

$$y_i^r = \begin{cases} argmax_k \boldsymbol{q}_i^r, & max(\boldsymbol{q}_i^r) \geq \epsilon \\ -1, & max(\boldsymbol{q}_i^r) < \epsilon \end{cases} \quad r \in \{a, b\} \quad (5)$$

where, ϵ is the confidence threshold, which is used to avoid false pseudo labels that are counterproductive. In order to ensure the consistency of pseudo-labels between positive pairs, they are checked for identical pseudo-labels as follows:

$$y_i = \begin{cases} y_i^a, & y_i^a = y_i^b \\ -1, & y_i^a \neq y_i^b \end{cases} \quad (6)$$

If the pseudo-labels of the positive pairs are different, pseudo-labels are considered untrusted. Then, the batch data is filtered through the pseudo-label to obtain a set of negative instances as follows:

$$S_i = \{j, y_i = -1 || y_j \neq y_i\} \quad (7)$$

To mitigate loss of information due to contrast loss, we map feature representation of samples to subspace through a nonlinear Multi-layer Perceptron (MLP) $g_I(\cdot)$ to obtain samples subspace representation $\boldsymbol{z}_i = g_I(\boldsymbol{h}_i)$, and calculate the contrast loss in the subspace as follows [16]:

$$l_i^a = -log\frac{exp(sim(\boldsymbol{z}_i^a, \boldsymbol{z}_i^b)/\tau)}{\sum_{j\in S_i} exp(sim(\boldsymbol{z}_i^a, \boldsymbol{z}_j^a)/\tau) + exp(sim(\boldsymbol{z}_i^a, \boldsymbol{z}_i^b)/\tau)} \quad (8)$$

where, τ is temperature parameter, and $sim(\cdot)$ is the similarity measure. In this paper, $sim(\cdot)$ is calculated by the normalized dot product between outputs as follows:

$$sim(\boldsymbol{z}_i^a, \boldsymbol{z}_i^b) = \frac{(\boldsymbol{z}_i^a)(\boldsymbol{z}_i^b)^T}{||\boldsymbol{z}_i^a||_2||\boldsymbol{z}_i^b||_2} \quad (9)$$

By Eq.(8), the respective contrast losses in all positive pairs in batch data are calculated and averaged as follows:

$$\mathcal{L}_I = \frac{1}{2M}\sum_{i=1}^{M}(l_i^a + l_i^b) \quad (10)$$

D. DYNAMICALLY ADJUSTED OBJECTIVE LOSS FUNCTION

Both the semantic-level clustering head and the instance-level constive head in our model are single-stage and end-to-end optimization processes, which joint train model by optimizing losses of two parts simultaneously. The total objective loss function consists of clustering loss and contrastive loss together is defined by:

$$\mathcal{L} = (\theta + (1-\lambda))\mathcal{L}_C + \sigma(\theta + \lambda)\mathcal{L}_I \quad (11)$$

where, θ is initial coefficient, σ is balance coefficient, and λ is adjustment function of clustering loss and contrastive loss. Contrast loss focuses more on learning sample features, while clustering loss focuses more on obtaining clustering results, so we hope that model first learns appropriate feature representations, and then smoothly transitions to clustering to get good clustering results. Therefore, the adjustment function is chosen as a monotonic decreasing function with a value interval of [0,1] as follows:

$$\lambda(l) = \frac{1}{2}\cos(\frac{l}{L}\pi) + \frac{1}{2} \quad (12)$$

where, l is the current iteration number and L is the expected total iteration number. At the beginning of training, the model is mainly trained by contrastive loss, learning better sample feature representation. As the training proceeds, the weight of contrastive loss decreases and the weight of cluster loss increases gradually by adjusting the function. The model gradually pays more attention to cluster learning until the training ends.

In the test, the original text $\boldsymbol{x}$ without augmentation is input model to obtain cluster soft-distribution $\boldsymbol{q}$ from clustering head. The samples are then divided into clusters with the highest probability of cluster soft distribution. By this way, the cluster division of each data in dataset can be obtained and concated together to form a predicted label vector $Y (Y \in \mathbb{R}^{N\times 1})$, where Y_i is the label predicted of i_{th} sample. In algorithm 1, a complete training and testing process is shown using pseudocode.

IV. EXPERIMENTS

In this section, we describe the experiment in this paper. Firstly, we introduce dataset used in experiment, the super parameters used in the model and the experimental evaluation indicators. Then, the validity of the model is verified by the

Algorithm 1 Clustering With Dynamic Adjustment for Contrastive Learning

Input: dataset X; iteration number L; batchsize M; cluster number K; pre-train encoder $f(\cdot)$; augmentations T_a, T_b; non-linear mlp $g_I(\cdot)$

Output: dataset X corresponding predicted label vector $Y(Y \in \mathbb{R}^{N\times 1})$

```
//training
initialization cluster center μ by K-Means
for l = 1 to L do
    randomly select M samples as mini-batch B = {x_i}_{i=1}^M from dataset X
    compute feature representations h_i^a, h_i^b by h_i^a = f(T_a(x_i)), h_i^b = f(T_b(x_b))
    compute cluster soft-assignments q^a, q^b, p^a, p^b by Eq.(1) and Eq.(3)
    comput cluster loss L_C by Eq.(4)
    comput subspace representations z_i by z_i = g_I(h_i)
    acquire peseudo-labels y_i by Eq.(5) and Eq.(6)
    filter negative sets S_i through Eq (7)
    comput contrastive loss L_I by Eq.(10)
    comput total loss L by Eq.(11)
    update f, g_I, μ by minimize L
end for
// test
define Y is predicted label vector
for x in X do
    compute feature representation h = f(x)
    compute cluster soft-assignments q by Eq.(1)
    compute cluster assignment by y = argmax_k q
    Y = [Y||y]
end for
```

TABLE 1. Dataset statistics.

Dataset	$\|V\|$	Documents		Clusters	
		N^D	Len	N^C	L/S
SearchSnippets	31K	12340	18	8	7
StackOverflow	15K	20000	8	20	1
Biomedical	19K	20000	13	20	1
AgNews	21K	8000	23	4	1
Tweet	5K	2472	8	89	249
GoogleNews-TS	20K	11109	28	152	143
GoogleNews-T	8K	11109	6	152	143
GoogleNews-S	18K	11109	22	152	143

comparison experiment with baseline model, and ablation experiment verifies that the methods presented in this paper can improve the clustering effect. Finally, the selection of the super parameters of the model is explained by the parameter experiment.

A. EXPERIMENTAL CONFIGURATIONS

1) DATASET

We evaluated the performance of DACL in short text clustering on eight benchmark datasets, and Table 1 summarizes the main statistics. Eight datasets are SearchSnippets [24], StackOverflow [25], Biomedical [25], AgNews [26], Tweet [27], GoogleNews-TS, GoogleNews-T, GoogleNews-S. GoogleNews-TS, GoogleNews-T, and GoogleNews-S are captured separately by extracting titles and summaries from the GoogleNews dataset [26]. To demonstrate validity of model and compare with baseline model, our model does not apply any pre-processing procedures on all datasets.

2) PARAMETER SETTINGS

We implement model in PyTorch [28] with the Sentence Transformer library [29]. We choose distilbert-base-nli-stsb-mean-tokens as the pre-training encoder. We use the ADAM optimizer [30], setting the batch size to 400, the total number of iterations to 2000, and the maximum text length to 32. The learning rate of encoder is 1e-5, and that of cluster head and contrast head is 1e-3. We use context augmentation for data augmentation and select Bert-base [31] and Roberta [32] to generate data augmentation pairs.

For cluster header, we use a linear layer of size $728 \times K$ to approximate the cluster centers, where K is the expected number of clusters. Set up $\alpha = 1$, but $\alpha = 10$ set in the Biomedical dataset. For the contrastive head, we use a two-layer linear layer and the ReLU activation function to make up a size of 728×128 MLP as a subspace mapping. Set up $\tau = 0.5$, $\epsilon = 0.3$. Set $\theta = 0.3$, $\sigma = 10$ in total loss.

3) EVALUATION METRICS

We evaluated our model using two widely used clustering measures, Normalized Mutual Information (NMI) and Accuracy (ACC). ACC denotes the degree of conformance when best mapping between predicting semantic clusters and ground-truth semantic clusters, and NMI denotes the normalized similarity measure between the two semantic clusters. The ACC and NMI formulas are as follows:

$$ACC = \frac{\sum_k^K \mathbb{1}\{l_i = map(c_i)\}}{N} \tag{13}$$

$$NMI = \frac{I(l;c)}{(H(l)+H(c))/2} \tag{14}$$

where, c denotes predicted semantic cluster, l denotes ground-truth semantic cluster, and $map(\cdot)$ denotes the best mapping between them, which is generally obtained by the Hungarian algorithm. $I(l;c)$ denotes the mutual information between c and l, and $H(\cdot)$ denotes information entropy. The value range of ACC and NMI is [0, 1], and higher values of metrics indicate better cluster performance.

B. COMPARISONS WITH BASELINE

To demonstrate the performance of our model on short text clustering. We compared eight representative baseline clustering methods on eight datasets, and the details of each baseline are as follows.

- **BoW** selects the 1500 most frequently occurring words in each dataset to form a 1500-dimensional feature vector, after removing the stop words. Each dimension is the

number of corresponding words in sample. Computing K-Means of eigenvectors to get clustering results.

- **TF-IDF** uses the same method as BoW to get a 1500-dimensional eigenvector. Each dimension is value of TF-IDF of corresponding word in sample.
- **K-Means** is applied to the feature vector obtained by the initialized encoder of the model we proposed.
- **DEC** [11] design new clustering loss and simultaneously learns feature representations and cluster assignments using deep neural networks.
- **STTC** [25] uses word2vec to embed the original text into the low dimensional feature space. Then it is input into the convolution neural network to learn the deep feature representation. Finally, the learned representation is clustered by K-means to obtain the optimal clustering.
- **Self-Train** [33] uses SIF to get the short text vector, and then uses autoencoder to reconstruct the short text vector. Finally, the model is fine-tuned by sharpening the cluster assignment probability.
- **HAC-HD** [26] applies hierarchical agglomerative clustering on top of a sparse pairwise similarity matrix obtained by zeroing-out similarity scores lower than a chosen threshold value.
- **SCCL** [14] uses contrast learning for representation learning and sharpened confidence distributions for clustering learning. Clustering is performed by jointly optimizing contrast loss and clustering loss.

In the comparative experiment, the results of our model are average of five repeated experiments, the results of baseline model from their paper. The comparison results are shown in Table 2. On most of the datasets, our model achieves the best performance compared to existing baseline models, especially the SCCL model that also uses a combination of contrast learning and clustering. Compared to SCCL, DACL achieves better performance by using dynamic adjustment of loss weights and negative instances filtering, and by pushing the cluster distribution of the augmented samples to its objective distribution of corresponding positive pairs in semantic-level clustering loss.

Because biomedical-related datasets have much less relevance with corpus of transformer pre-trained models, and Self-Train pre-trained word embeddings on large biomedical corpus, the model is less effective on the Biomedical dataset than the Self-Train model. Because the GoogleNews and Tweet datasets have fewer training samples and more clusters, contrastive learning requires a large number of training samples, and the clustering loss obtained by Student's t-distribution will have performance degradation when the number of clusters is large, HAC-SD model achieves better performance by applying agglomerative clustering on carefully selected pairwise similarities of pre-processed data.

Our model uses Bert as encoder and performs one-stage and end-to-end training, so the time complexity of DACL is $O(M^2N)$, where N is iterations and M is the text length. In baseline model, The HAC-SD model uses agglomerative clustering method with time complexity of $O(MN^2 \log(N/K))$, where K is the number of clusters. The DEC and Self-Train model use autoencoder as encoder, with time complexity of $O(MN)$. The STTC model uses convolution neural networks for encoding and SCCL model uses Bert for encoding, so their time complexity is $O(M^2N)$. Because of the large number of samples and the short length of short text dataset, N is much larger than M. Therefore, except for HAC-SD, the cost of training the model is roughly equivalent.

C. QUALITATIVE STUDY

In this section, we conduct experiments on the SearchSnippets dataset to analyze the evolution of data in feature space during training. To understand how the model is trained by optimizing loss and obtaining the cluster distribution of samples. We select the distribution of samples at different training moments in the feature space and use T-SNE to reduce dimension. The results are shown in Figure 3, the initial features of samples are mixed, and the distribution of features is more reasonable with training of contrastive learning. Then, due to the introduction of the cluster loss, the sample approaches the cluster center.

We evaluate the intra cluster distance in the process of model training, and the results are shown in Figure 4. When the training focus of model is on contrastive learning, the samples are scattered in the feature space, and as the training focus transits to distance loss, the samples are gradually concentrated in the cluster center.

D. ABLATION EXPERIMENT

In this section. To verify the positive impact of our proposed method on clustering, we performed ablation experiments on the SeachSnippets dataset. We first assessed the use of negative case screening, then assessed the method of joint learning both contrastive loss and clustering loss. As shown in Table 3, the use of negative case filtering can improve the ACC and NMI of clustering results when other settings are the same. Compared with fixed weights and sequential training, the dynamic adjustment of loss weights achieves the best results.

E. PARAMETER EXPERIMENT

In this section, we will evaluate the impact of hyperparameters on model performance. On the StackOverflow dataset, we conduct experiments on loss initial coefficients θ, loss adjustment functions λ, confidence thresholds respectively σ and initial cluster center μ.

1) INITIAL LOSS COEFFICIENT

The initial loss coefficient parameter is initial weight of contrastive loss and clustering loss in total loss. The smaller this parameter, the smaller the fixed weight of the loss in training. The result is shown in Figure 5, this parameter achieves the best result at 0.5. Low initial coefficient of loss results in slow updating of cluster head during earlier training, and high

TABLE 2. Experimental results. Our results are average of five repeated experiments.

Model	Searchnippets		StackOverflow		Biomedical		AgNews	
	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
BoW	24.3	9.3	18.5	14.0	14.3	9.2	27.6	2.6
TF-IDF	31.5	19.2	58.4	58.7	28.3	23.2	34.5	11.9
K-Means	59.0	36.4	60.8	52.3	39.8	32.7	83.9	59.2
DEC	76.9	64.9	74.7	75.3	41.6	37.7	-	-
STCC	77.0	63.2	59.8	54.8	43.6	38.1	-	-
Self-Train	77.1	56.7	64.8	64.8	**54.8**	**47.1**	-	-
HAC-SD	82.7	63.8	64.8	59.5	40.1	33.5	82.8	54.6
SCCL	85.2	71.1	75.5	74.5	46.2	41.5	88.2	68.2
DACL	**86.1**	**73.6**	**77.5**	**76.0**	48.6	40.3	**88.6**	**69.0**
Model	Tweet		GoogleNews-TS		GoogleNews-T		GoogleNews-S	
	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
BoW	49.7	73.6	57.5	81.9	49.8	73.2	49.0	73.5
TF-IDF	57.0	80.7	68.0	88.9	58.9	79.3	61.9	83.0
K-Means	51.7	79.0	56.0	78.4	62.2	83.3	67.8	87.5
DEC	-	-	-	-	-	-	-	-
STCC	-	-	-	-	-	-	-	-
Self-Train	-	-	-	-	-	-	-	-
HAC-SD	**89.6**	85.2	85.8	88.0	**81.8**	84.2	80.6	83.5
SCCL	78.2	89.2	**89.8**	**94.9**	75.8	88.3	83.1	90.4
DACL	82.0	**90.6**	89.6	94.4	80.5	**89.8**	**84.0**	**91.7**

TABLE 3. Ablation experiments.

Negative instances filter	Mathod of joint learning	ACC	NMI
Unused	$\text{seq}(\mathcal{L}_I, \mathcal{L}_C)$	78.0	59.9
	$\mathcal{L}_C + \sigma\mathcal{L}_\mathcal{I}$	85.3	72.0
	$(\theta + (1-\lambda))\mathcal{L}_C + \sigma(\theta+\lambda)\mathcal{L}_i$	85.8	72.6
Used	$\text{seq}(\mathcal{L}_I, \mathcal{L}_C)$	78.5	60.7
	$\mathcal{L}_C + \sigma\mathcal{L}_\mathcal{I}$	85.4	72.4
	$(\theta + (1-\lambda))\mathcal{L}_C + \sigma(\theta+\lambda)\mathcal{L}_i$	**86.1**	**73.6**

initial coefficient results in high weight of contrastive loss during later training.

2) LOSS ADJUSTMENT FUNCTION

The loss adjustment function is a function whose interval is in [0,1], which decreases monotonically with the number of iterations. It is used to achieve a smooth transition from contrastive loss to clustering loss. As shown in Figure 6, we compare four adjustment functions: convex, linear, concave, and composite as follows [34]:

$$\begin{cases} \lambda(l) = \cos(\dfrac{l}{L} \times \dfrac{\pi}{2}) \\ \lambda(l) = 1 - \dfrac{l}{L} \\ \lambda(l) = \beta^l \\ \lambda(l) = \dfrac{1}{2}\cos(\dfrac{l}{L}\pi) + \dfrac{1}{2} \end{cases} \tag{15}$$

where, β is base of the exponential function, which is fixed as 0.995 in this paper. The result is shown in Table 4, Composite function achieved the highest ACC and NMI and achieved the best performance. Compared with other transfer functions,

TABLE 4. Loss adjustment function.

Loss adjustment function	ACC	NMI
$1 - l/L$	76.2	75.3
β^l	76.5	74.0
$\cos(l/L \times \pi/2)$	77.1	75.2
$1/2 * \cos(l/L \times \pi) + 1/2$	**77.4**	**75.5**

the transfer rate of the composite function is from slow to fast and then from fast to slow, which ensures that the model can not only learn the characteristic representation of the samples but also obtain the clustering results with high confidence.

3) CONFIDENCE THRESHOLD

The confidence threshold is parameter that is used to determine cluster assignment when negative example filtering for contrastive learning. Data whose cluster assignment probability is greater than this parameter is assigned to a cluster. As shown in Figure 7, the best result is obtained at 0.8. Low confidence thresholds result in data being wrongly assigned to a cluster, and high thresholds result in too few data being filtered.

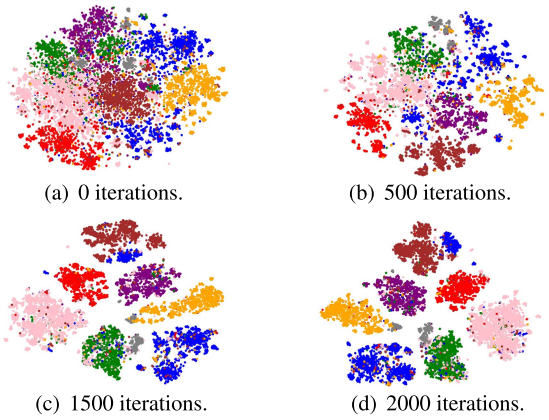

FIGURE 3. Evolution of samples in feature space during training on SearchSnippets. Each point represents each sample, and each color represents each true semantic cluster.

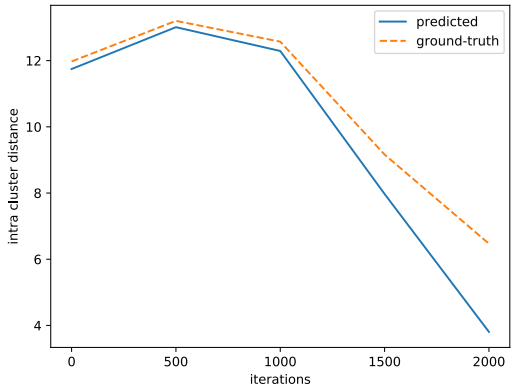

FIGURE 4. Intra cluster distance.

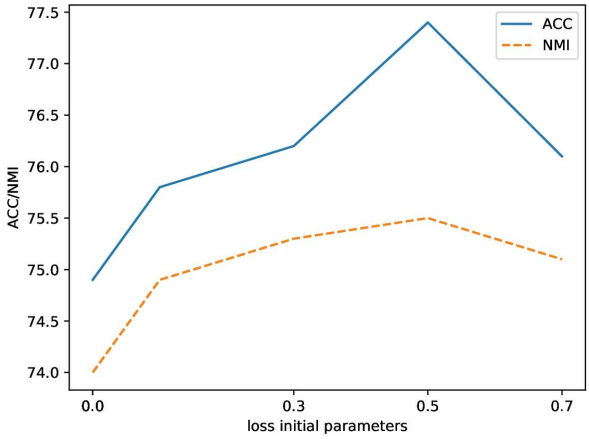

FIGURE 5. Initial loss coefficient.

4) INITIAL CLUSTER CENTER

The initial cluster center is the cluster semantic center set before model training begins. The initial cluster center is represented by a linear layer in the cluster header. With clustering loss optimization, on the one hand, the samples in the cluster are closer to cluster center, on the other hand, cluster center can be updated with the update of the spatial distribution of samples. We compare three initial clustering methods: random, BIRCH, and K-Means. As shown in Table 5, K-Means achieved best cluster results.

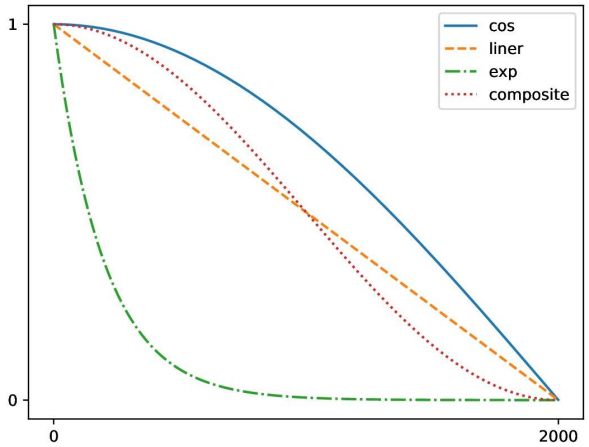

FIGURE 6. Loss adjustment function

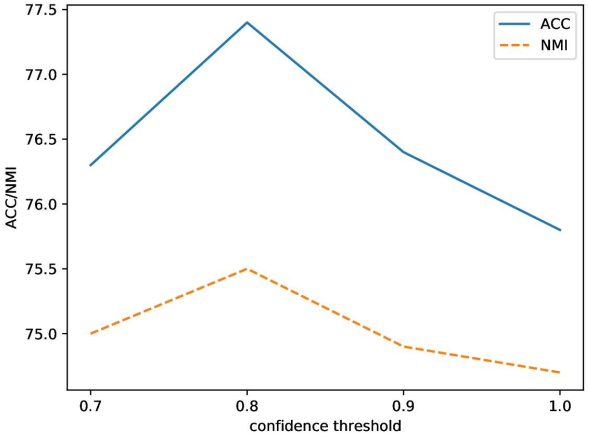

FIGURE 7. Confidence threshold.

TABLE 5. Initial cluster center.

Cluster method	ACC	NMI
Random	23.4	24.4
BIRCH	76.0	74.5
K-Means	**77.3**	**75.4**

V. CONCLUSION AND FUTURE WORK

In this paper, we propose a new short text clustering method that dynamically combines contrastive learning with unsupervised clustering to solve the problem of inconsistency objective between contrastive learning and clustering. We evaluated the model on eight datasets and achieved better or comparable results than state-of-the-art methods. In addition, the Validity of the model was verified by ablation experiment. The verification results show that dynamic adjustment of loss of contrastive learning and clustering and negative instances filter are beneficial to the improvement of clustering result, which can more effectively solve categorize problem of a large number of short text on the Internet. However, as with most related work on clustering, this paper relies on K-Means to initialize cluster centers on the entire dataset. In large-scale and online scenes, facing the large number of text streams currently being generated, it is not possible to handle the issues of feature shift and clustering initialization. In future work,

we will try to solve the problem of data stream clustering by incremental clustering.

REFERENCES

[1] D. M. Boyd and N. B. Ellison, "Social network sites: Definition, history, and scholarship," *J. Comput.-Mediated Commun.*, vol. 13, no. 1, pp. 210–230, Oct. 2007.
[2] S. Z. Selim and M. A. Ismail, "*K*-means-type algorithms: A generalized convergence theorem and characterization of local optimality," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-6, no. 1, pp. 81–87, Jan. 1984.
[3] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Stat. Soc., B, Methodol.*, vol. 39, no. 1, pp. 1–22, 1977.
[4] X. Yan, J. Guo, Y. Lan, and X. Cheng, "A biterm topic model for short texts," in *Proc. 22nd Int. Conf. World Wide Web (WWW)*, 2013, pp. 1445–1456.
[5] E. Min, X. Guo, Q. Liu, G. Zhang, J. Cui, and J. Long, "A survey of clustering with deep learning: From the perspective of network architecture," *IEEE Access*, vol. 6, pp. 39501–39514, 2018.
[6] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon, "A survey on contrastive self-supervised learning," *Technologies*, vol. 9, no. 1, p. 2, Dec. 2020.
[7] P. Baldi, "Autoencoders, unsupervised learning, and deep architectures," in *Proc. ICML Workshop Unsupervised Transf. Learn.*, in JMLR Workshop and Conference Proceedings, 2012, pp. 37–49.
[8] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Process.*, vol. 35, no. 1, pp. 53–65, Jan. 2017.
[9] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 9912–9924.
[10] B. Yang, X. Fu, N. D. Sidiropoulos, and M. Hong, "Towards K-means-friendly spaces: Simultaneous deep learning and clustering," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 3861–3870.
[11] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 478–487.
[12] J. Chang, L. Wang, G. Meng, S. Xiang, and C. Pan, "Deep adaptive image clustering," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5879–5887.
[13] J. Wu, K. Long, F. Wang, C. Qian, C. Li, Z. Lin, and H. Zha, "Deep comprehensive correlation mining for image clustering," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8150–8159.
[14] D. Zhang, F. Nan, X. Wei, S.-W. Li, H. Zhu, K. McKeown, R. Nallapati, A. O. Arnold, and B. Xiang, "Supporting clustering with contrastive learning," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2021, pp. 5419–5430. [Online]. Available: https://aclanthology.org/2021.naacl-main.427
[15] X. Ji, A. Vedaldi, and J. Henriques, "Invariant information clustering for unsupervised image classification and segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9865–9874.
[16] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.
[17] I. Misra and L. van der Maaten, "Self-supervised learning of pretext-invariant representations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6707–6717.
[18] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9729–9738.
[19] T. T. Cai, J. Frankle, D. J. Schwab, and A. S. Morcos, "Are all negatives created equal in contrastive instance discrimination?" 2020, *arXiv:2010.06682*.
[20] Y. Li, P. Hu, Z. Liu, D. Peng, J. T. Zhou, and X. Peng, "Contrastive clustering," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2021, pp. 8547–8555.
[21] S. Kobayashi, "Contextual augmentation: Data augmentation by words with paradigmatic relations," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.* New Orleans, LA, USA: Assoc. Comput. Linguistics, 2018, pp. 452–457. [Online]. Available: https://aclanthology.org/N18-2072
[22] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.
[23] A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, and T. Brox, "Discriminative unsupervised feature learning with convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 1–9.
[24] X.-H. Phan, L.-M. Nguyen, and S. Horiguchi, "Learning to classify short and sparse text & web with hidden topics from large-scale data collections," in *Proc. 17th Int. Conf. World Wide Web (WWW)*, 2008, pp. 91–100.
[25] J. Xu, B. Xu, P. Wang, S. Zheng, G. Tian, and J. Zhao, "Self-taught convolutional neural networks for short text clustering," *Neural Netw.*, vol. 88, pp. 22–31, Apr. 2017.
[26] M. R. H. Rakib, N. Zeh, M. Jankowska, and E. Milios, "Enhancement of short text clustering by iterative classification," in *Proc. Int. Conf. Appl. Natural Lang. Inf. Syst.* Cham, Switzerland: Springer, 2020, pp. 105–117.
[27] J. Yin and J. Wang, "A model-based approach for text clustering with outlier detection," in *Proc. IEEE 32nd Int. Conf. Data Eng. (ICDE)*, May 2016, pp. 625–636.
[28] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *Proc. NIPS Workshop Autodiff*, 2017, pp. 1–4. [Online]. Available: https://openreview.net/forum?id=BJJsrmfCZ
[29] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*. Hong Kong, 2019, pp. 3982–3992. [Online]. Available: https://aclanthology.org/D19-1410
[30] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[31] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 7871–7880. [Online]. Available: https://aclanthology.org/2020.acl-main.703
[32] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.
[33] A. Hadifar, L. Sterckx, T. Demeester, and C. Develder, "A self-training approach for short text clustering," in *Proc. 4th Workshop Represent. Learn. NLP (RepL4NLP)*, 2019, pp. 194–199.
[34] Y. Wang, W. Gan, J. Yang, W. Wu, and J. Yan, "Dynamic curriculum learning for imbalanced data classification," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5017–5026.

RUIHUI LI was born in 1997. He received the B.E. degree from the Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming, China, in 2019, where he is currently pursuing the M.E. degree. His research interests include natural language processing and deep cluster.

HONGBIN WANG was born in 1983. He received the Ph.D. degree in computer science from Jilin University, Changchun, China, in 2013. He is an Associate Professor with the Faculty of Information Engineering and Automation, Kunming University of Science and Technology. His current research interests include intelligent information systems, natural language processing, and data analysis.