Improved Graph Contrastive Learning for Short Text Classification

Yonghao Liu[1], Lan Huang[1], Fausto Giunchiglia[2], Xiaoyue Feng[1*], Renchu Guan[1*]

[1]Key Laboratory of Symbolic Computation and Knowledge Engineering of the Ministry of Education, College of Computer Science and Technology, Jilin University
[2]University of Trento
yonghao20@mails.jlu.edu.cn, {huanglan, fengxy, guanrenchu}@jlu.edu.cn, fausto.giunchiglia@unitn.it

Abstract

Text classification occupies an important role in natural language processing and has many applications in real life. Short text classification, as one of its subtopics, has attracted increasing interest from researchers since it is more challenging due to its semantic sparsity and insufficient labeled data. Recent studies attempt to combine graph learning and contrastive learning to alleviate the above problems in short text classification. Despite their fruitful success, there are still several inherent limitations. First, the generation of augmented views may disrupt the semantic structure within the text and introduce negative effects due to noise permutation. Second, they ignore the clustering-friendly features in unlabeled data and fail to further utilize the prior information in few valuable labeled data. To this end, we propose a novel model that utilizes improved **G**raph contrast**I**ve learning for short text classi**F**ica**T**ion (**GIFT**). Specifically, we construct a heterogeneous graph containing several component graphs by mining from an internal corpus and introducing an external knowledge graph. Then, we use singular value decomposition to generate augmented views for graph contrastive learning. Moreover, we employ constrained k-means on labeled texts to learn clustering-friendly features, which facilitate cluster-oriented contrastive learning and assist in obtaining better category boundaries. Extensive experimental results show that GIFT significantly outperforms previous state-of-the-art methods. Our code can be found in https://github.com/KEAML-JLU/GIFT.

Introduction

Text classification aims to assign texts to predefined categories, which is a classic problem in natural language processing (Minaee et al. 2021). Most models are designed for regular texts, which include rich contextual information and sufficient labeled training data. However, short texts are ubiquitous in our daily life, such as tweets, news seeds, and search snippets. When these models are directly applied to short texts, they generally obtain unsatisfactory performance, suffering from limited contextual information and severe label scarcity problems (Wang et al. 2021). Compared to regular texts, the length of short texts is small, often containing only one or few words, which increases the difficulty of understanding their meaning correctly (Phan, Nguyen, and Horiguchi 2008). Moreover, the rise of the internet has led to an exponential increase in the volume of short texts. Unlabeled short text data has become significantly more abundant compared to labeled data (Hu et al. 2019). Therefore, short text classification (STC), as a highly challenging task that attracts tremendous attention from researchers, has a wide range of practical applications, such as news classification (Chen et al. 2019), sentiment analysis (Yao, Mao, and Luo 2019), and social media analysis (Liu et al. 2021, 2023a,b). Recently, some studies (Su et al. 2022) have attempted to integrate graph neural networks (GNNs) with contrastive learning (CL) for solving STC tasks, with promising results. In these approaches, a corpus-level graph is constructed, incorporating latent topics, words, or entities as nodes. GNNs are then performed to explicitly model the semantic relationships between nodes, enriching the original short text information with such auxiliary knowledge. Meanwhile, CL has the ability to extract self-supervised signals from sufficient unlabeled data to facilitate the model to learn superior representations.

Despite their effectiveness, some limitations still remain. First, when generating augmented views for CL, there are two types of approaches. *On the one hand*, they use traditional text augmentation methods such as random word deletion or random noise injection. However, these methods may lead to irreversible semantic distortion and information loss (Zhang et al. 2021). For example, removing the word "not" from the movie review sentence "this film is not funny" would completely change its meaning, leading to a misleading label. *On the other hand*, they perform edge/node perturbation on the built text graph for graph augmentation operations, which inevitably results in noise (Yu et al. 2022). For example, removing an essential semantic edge could considerably change the sentence's meaning and enable the augmented graph to share little learnable invariance with the original graph, thus misleading the model's whole learning process.

Second, they follow the classical instance discrimination CL paradigm, which maximizes the mutual information between positive sentence pairs from the same source while pushing negative counterparts from other instances away (Chen et al. 2020). However, this paradigm could lead to many generated negative pairs sharing similar semantics but

*Corresponding author.

being forced apart in the embedding space (Li et al. 2021), which negatively impacts the representation learning. To achieve better generalization, it is crucial to capture similarities within a class of samples and contrast them with samples from another class. In other words, the clustering-friendly characteristics contained in the dataset should be emphasized. The challenge lies in enabling unlabeled short texts with similar semantics to share the same weak labels, facilitating comparisons across different categories. While performing k-means clustering on the unlabeled data might seem like a natural approach, it fails to leverage prior knowledge about the distribution of hidden class labels obtained from limited labeled data. Importantly, this prior knowledge can guide the clustering process by biasing it towards exploring favorable regions in the search space and reducing the risk of converging to a suboptimal solution. Therefore, it is necessary to leverage these limited labeled data to further guide the clustering process.

To solve the aforementioned limitations, we propose a novel model called GIFT for STC tasks. Specifically, we first build a corpus-level heterogeneous graph that contains three component graphs, including a word graph, an entity graph, and a part-of-speech (POS) tag graph. These component graphs incorporate rich semantic and syntactic information to alleviate the semantic sparsity problem. Next, to preserve the important semantic structure of the original text, we perform singular value decomposition (SVD) on the term-document (TD) matrix, to obtain the reconstructed TD matrix for generating the augmented view of each component graph. The approach is inspired by latent semantic indexing (LSI) (Deerwester et al. 1990), which assumes that underlying semantic structures exist within the TD matrix. By performing dimension reduction via SVD, we retain useful information in the matrix while removing noise, such as word misuse or the presence of unrelated words. We adopt truncated SVD to obtain a low-rank approximation of the TD matrix, such that global TD co-occurrence signals (*i.e.*, similar texts with close semantic structures that share many keyword-related words.) can be captured and injected into the learning process of CL. Subsequently, we employ the constrained seed k-means algorithm to assign weak labels for unlabeled texts. This algorithm utilizes seed samples with the same label to form initial clusters, and the mean vector within each cluster serves as the initial centroid. After clustering, these samples with weak labels are used to perform supervised CL.

Overall, the main contributions of this work are listed below.

- We propose a novel model, namely GIFT, for STC tasks, which can learn better short text representations and solve the challenges of existing models.
- We perform SVD to obtain the augmented views of component graphs used for CL. Moreover, we utilize the constrained seed k-means algorithm to assign weak labels to unlabeled texts for integrating clustering information into cluster-oriented CL.
- We conduct extensive experiments on several benchmark datasets, and the results demonstrate our proposed model significantly outperforms other competitive models.

Related Work

Short Text Classification. STC poses unique challenges due to the limited length and lack of strict syntactic structure in short texts (Wang et al. 2017). Previous methods attempt to inject more additional information, such as latent topics extracted from external corpus (Zeng et al. 2018) and entity information residing in the knowledge base (Chen et al. 2019), to enrich their semantics. However, these methods fail to deliver satisfactory performance on the STC task because they merely alleviate the semantic sparsity problem and do not take measures for insufficient labeled data. Therefore, some GNN-based models (Hu et al. 2019; Ye et al. 2020; Wang et al. 2021) are proposed for this task and achieve improved performance. These models represent texts as graphs constructed based on local features such as shared words or phrases within a corpus. Label information is propagated through message passing on the built graph. Inspired by the success of CL in unsupervised representation learning, recent studies (Su et al. 2022) explore the potential of CL based on GNNs to leverage the self-supervised signals present in the unlabeled data, aiding in extracting useful features. However, their effectiveness is heavily relies on the generated contrastive views, and the way the views are generated can easily lead to incorrect self-supervised signals that misguide model learning.

Contrastive Learning. CL enables learning meaningful representations from large-scale unlabeled data, which has been proven successful in various fields, including computer vision (Chen et al. 2020) and natural language processing (Gao, Yao, and Chen 2021). The core concept of CL is to maximize the agreement of the positive pairs formed by the original instance and its corresponding augmented instance, while minimizing the agreement with the negative pairs formed by other instances. The initial CL models (He et al. 2020; Chen and He 2021) primarily focus on instance discrimination in an unsupervised manner, and subsequent models (Khosla et al. 2020; Gunel et al. 2021) propose the fully supervised CL paradigm by incorporating label information, which is based on category discrimination. Some studies (Caron et al. 2020; Li et al. 2021) about image classification tasks introduce the concept of prototypes, which involves forcing the embedding of instances closer to their corresponding prototypes in the embedding space, while pushing them away from other prototypes.

Low-rankness in Data Mining. Low-rankness is a common property of matrices used to describe the correlation among rows or columns (Wu et al. 2022). In data mining, low-rankness is often associated with matrix decomposition techniques such as SVD. By decomposing a high-rank matrix into an approximate low-rank matrix, we can extract latent features or structures, thereby achieving data dimension reduction, as well as reducing noise and redundant information (Hansen 1987). The property of low-rankness has been leveraged in various specific tasks within data mining. For example, in (Entezari et al. 2020), it adopts low-rank approximation of the adjacency matrix to discard higher-order components of the underlying graph for graph adversarial attack. In (Cai et al. 2023), it performs low-rank approxi-

mation operations on the user-item interaction matrix and injects global collaborative context to guide graph augmentation for graph-based recommendations.

Method

In this section, we provide detailed descriptions of the proposed model GIFT. For ease of understanding, we show the overall framework in Fig. 1.

Graph Construction

To maximize the usage of valid information within short texts, such as semantic and syntactic information, and auxiliary knowledge from external knowledge bases, such as entity information, we construct a corpus-level heterogeneous graph consisting of *a word graph*, *an entity graph*, and *a POS tag graph*. The detailed construction process is as follows.

The word graph $\mathcal{G}_w = \{\mathcal{V}_w, \mathbf{X}_w, \mathbf{A}_w\}$ is composed of words that constitute the short text, which contains necessary semantic information. $\mathcal{V}_w$ is the set of words and $\mathbf{X}_w \in \mathbb{R}^{|\mathcal{V}_w| \times f_w}$ is the word embeddings initialized by pre-trained GloVe word vectors. $\mathbf{A}_w \in \mathbb{R}^{|\mathcal{V}_w| \times |\mathcal{V}_w|}$ is the adjacency matrix based on the co-occurrence statistics of words within the text, where each value is determined by point-wise mutual information (PMI), *i.e.*, $[\mathbf{A}_w]_{ij} = \max(\text{PMI}(\mathcal{V}_{w,i}, \mathcal{V}_{w,j}), 0)$.

The entity graph $\mathcal{G}_e = \{\mathcal{V}_e, \mathbf{X}_e, \mathbf{A}_e\}$ is constructed by the entities residing in knowledge graphs, which can offer auxiliary information for texts. $\mathcal{V}_e$ denotes the set of entities and $\mathbf{X}_e \in \mathbb{R}^{|\mathcal{V}_e| \times f_e}$ denotes the entity embeddings initialized by TransE (Bordes et al. 2013). $\mathbf{A}_e \in \mathbb{R}^{|\mathcal{V}_e| \times |\mathcal{V}_e|}$ is determined by the cosine similarity of each entity pair, *i.e.*, $[\mathbf{A}_e]_{ij} = \max(\cos(\mathcal{V}_{e,i}, \mathcal{V}_{e,j}), 0)$.

The POS tag graph $\mathcal{G}_p = \{\mathcal{V}_p, \mathbf{X}_p, \mathbf{A}_p\}$ consists of POS tags such as nouns and verbs for words, which specify the syntactic roles of words to help eliminate ambiguity. $\mathcal{V}_p$ is the POS tag set and $\mathbf{X}_p \in \mathbb{R}^{|\mathcal{V}_p| \times f_p}$ is the tag node features initialized by one-hot vectors. $\mathbf{A}_p \in \mathbb{R}^{|\mathcal{V}_p| \times |\mathcal{V}_p|}$ is the corresponding adjacency matrix also calculated by PMI.

Text Representation Learning

After building the graphs, we encode them with GNNs to simultaneously leverage the topology and feature information. Concretely, we adopt graph convolutional networks (GCNs), which are defined as follows:

$$\mathbf{H}^{(\ell+1)} = \sigma(\hat{\mathbf{D}}^{-\frac{1}{2}}\hat{\mathbf{A}}\hat{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{(\ell)}\mathbf{W}^{(\ell)}) \tag{1}$$

where $\mathbf{H}^{(\ell)}$ denotes the node embeddings at layer ℓ and $\mathbf{H}^{(0)} = \mathbf{X}$ indicates the initialized features. $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ is the adjacency matrix with self-loops and $\hat{\mathbf{D}}_{ii} = \sum_j \hat{\mathbf{A}}_{ij}$ is the degree matrix. $\mathbf{W}$ and $\sigma(\cdot)$ denote the trainable parameter and activation function, respectively.

Given three types of graphs $\mathcal{G}_\pi = \{\mathcal{V}_\pi, \mathbf{X}_\pi, \mathbf{A}_\pi\}$, $\pi \in \{w, e, p\}$, we can use Eq. 1 to obtain the updated node embeddings $\mathbf{H}_\pi \in \mathbb{R}^{|\mathcal{V}_\pi| \times f_\pi}$, which take advantage of interactions between nodes.

Subsequently, to derive text embeddings, we construct text-specific matrices (*i.e.*, TD matrices) for each type of node (*i.e.*, word, entity, and POS tag) to establish connections between the text and node. For words or POS tags, we set $\mathbf{M}_\pi \in \mathbb{R}^{N \times |\mathcal{V}_\pi|}, \pi \in \{w, p\}$ to the TF-IDF value between each text and word or POS tag in the corpus, where N is the number of texts. For entities, we let $\mathbf{M}_e \in \mathbb{R}^{N \times |\mathcal{V}_e|}$, where $\mathbf{M}_{e,ij} = 1$ if the i-th text contains the j-th entity, and 0 otherwise. Then, we adopt an information aggregation operation as follows:

$$\begin{aligned} \mathbf{Z}_\pi &= \mathbf{M}_\pi \mathbf{H}_\pi, \pi \in \{w, e, p\} \\ \mathbf{Z}_{\text{org}} &= \mathbf{Z}_w || \mathbf{Z}_e || \mathbf{Z}_p \end{aligned} \tag{2}$$

where $\mathbf{Z}_\pi \in \mathbb{R}^{N \times f_\pi}$ indicates the text-specific features with respect to nodes of type π. The text representation $\mathbf{Z}_{\text{org}}$ is obtained by concatenating three text-relevant features.

Improved Graph Contrastive Learning

As mentioned earlier, existing methods for generating augmented text views often suffer from the loss of important semantics or introduce noise, thereby misleading the CL learning process. To mitigate this issue, we perform SVD matrix decomposition on the TD matrix of each node type for dimensionality reduction and denoising, which can be mathematically expressed as:

$$\mathbf{M}_\pi = \mathbf{U}_\pi \Sigma_\pi \mathbf{V}_\pi^\top, \pi \in \{w, e, p\} \tag{3}$$

where $\mathbf{U}_\pi \in \mathbb{R}^{N \times N}$ and $\mathbf{V}_\pi \in \mathbb{R}^{|\mathcal{V}_\pi| \times |\mathcal{V}_\pi|}$ are orthogonal matrices in which the columns of $\mathbf{U}_\pi$ and $\mathbf{V}_\pi$ are left singular vectors and right singular vectors, respectively. $\Sigma_\pi \in \mathbb{R}^{N \times |\mathcal{V}_\pi|}$ is a real-valued diagonal matrix that stores the singular values in descending order. The singular values indicate the contribution of the corresponding principal components to the original matrix. Typically, the sum of the leading 10% or even 1% of the singular values accounts for over 99% of the total sum of the singular values. Therefore, we can use truncated SVD to preserve the largest r singular values to obtain a low-rank approximation of the original matrix, which is computed as follows:

$$\mathbf{M}_{\pi,r} = \mathbf{U}_{\pi,r} \Sigma_{\pi,r} \mathbf{V}_{\pi,r}^\top, \pi \in \{w, e, p\} \tag{4}$$

where $\mathbf{M}_{\pi,r}$ is the rank-r approximation of $\mathbf{M}_\pi$. $\mathbf{U}_{\pi,r} \in \mathbb{R}^{N \times r}$ and $\mathbf{V}_{\pi,r} \in \mathbb{R}^{|\mathcal{V}_\pi| \times r}$ are matrices composed of the top r singular vectors and $\Sigma_{\pi,r} \in \mathbb{R}^{r \times r}$ is the diagonal matrix containing the largest r singular values.

By performing SVD on the original TD matrix, we achieve two objectives. First, we enhance the prominent and credible TD co-occurrence signals for text representation while reducing the noisy co-occurrence signals. Second, we address the issue of semantic ambiguity caused by the limited local TD co-occurrence signals by incorporating global TD co-occurrence signals from each TD pair. Then, we apply an information aggregation mechanism on the low-rank approximation matrix for generating the augmented views of texts, which can be expressed as follows:

$$\begin{aligned} \mathbf{Z}_{\pi,r} &= \mathbf{M}_{\pi,r} \mathbf{H}_\pi, \pi \in \{w, e, p\} \\ \mathbf{Z}_{\text{aug}} &= \mathbf{Z}_{w,r} || \mathbf{Z}_{e,r} || \mathbf{Z}_{p,r} \end{aligned} \tag{5}$$

For ease of presentation, we denote $\mathbf{Z} = \mathbf{Z}_{\text{org}} || \mathbf{Z}_{\text{aug}}$. After obtaining two views of texts, we first introduce a projection head $\Phi(\cdot)$ for mapping text representations into the

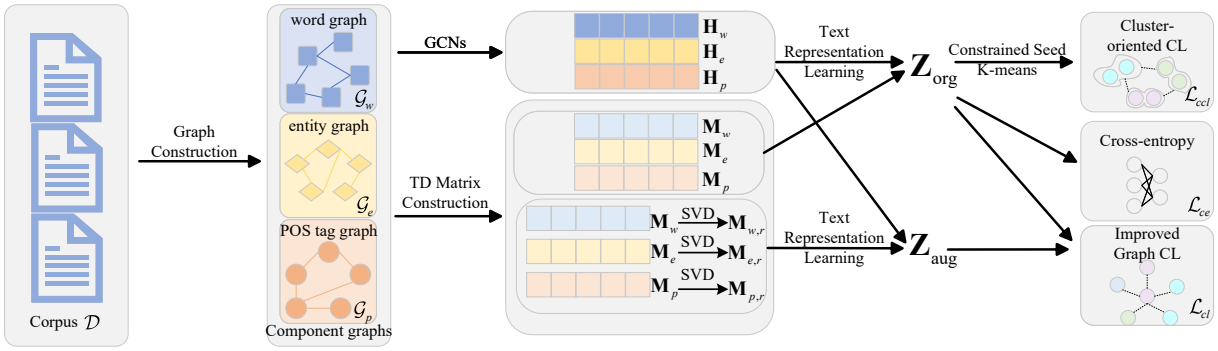


Figure 1: The overall architecture of our model. First, we construct a heterogeneous graph consisting of three component graphs $\mathcal{G}_w, \mathcal{G}_e$ and $\mathcal{G}_p$, then perform GCNs on each component graph to obtain the updated node embeddings $\mathbf{H}_w, \mathbf{H}_e$, and $\mathbf{H}_p$, respectively. Meanwhile, we build the text-specific TD matrices $\mathbf{M}_w, \mathbf{M}_e$, and $\mathbf{M}_p$, and perform SVD on them to obtain the low-rank approximate matrices $\mathbf{M}_{w,r}, \mathbf{M}_{e,r}$, and $\mathbf{M}_{p,r}$. Then, we obtain $\mathbf{Z}_{\text{org}}$ and $\mathbf{Z}_{\text{aug}}$ through the text representation learning module. We utilize $\mathbf{Z}_{\text{org}}$ and $\mathbf{Z}_{\text{aug}}$ for improved graph CL, and perform cross entropy loss and cluster-oriented CL with constrained seed k-means on $\mathbf{Z}_{\text{org}}$.

hidden space where the contrastive loss is applied, and then normalize hidden representations into unit form. *i.e.*, $\mathbf{P} = \text{norm}(\Phi(\mathbf{Z}))$. The mathematical expression of contrastive loss can be denoted as:

$$\begin{aligned} \mathcal{L}_i &= -\log \frac{\exp((\mathbf{P}_i \cdot \mathbf{P}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{I}_{k \neq i} \exp((\mathbf{P}_i \cdot \mathbf{P}_k)/\tau)} \\ \mathcal{L}_{cl} &= \frac{1}{2N} \sum_{i=1}^{2N} \mathcal{L}_i \end{aligned} \tag{6}$$

where $(\mathbf{P}_i, \mathbf{P}_j)$ is the defined positive pair in which i and j denote the indices of the representations of the same text. $\mathbb{I}$ is the indicator function set to 1 if $k \neq 1$ and 0 otherwise. τ and $\cdot$ are the temperature parameter and dot product operator.

By conducting Eq. 6, the model can bring the positive sample pairs closer together in the feature space while pushing the negative sample pairs further apart, thereby inducing more discriminative embeddings for downstream tasks.

Cluster-oriented Contrastive Learning

However, solely relying on the above CL paradigm is insufficient because it ignores clustering-friendly information involved in the data and treats each instance independently, leading to false positive pairs with similar semantics. In other word, the only positive pair consists of augmented views generated from the same instance, while other instances with similar high-level semantics are misjudged as negative ones. The wrong negative instances are pushed away during subsequent CL. To alleviate this issue while explicitly leveraging the prior knowledge brought by the few labeled texts, we propose assigning weak labels to numerous unlabeled texts in the corpus using the constrained seed k-means algorithm, to explore clustering-friendly features that can assist the model in obtaining more discriminative class boundaries. Specifically, given the corpus $\mathcal{D} = \{d_1, \cdots, d_N\}$, after the text representation learning module, we can obtain the corresponding text embeddings $\mathbf{Z}_{\text{org}}$. We denote the labeled text set as $\mathcal{D}_{lab} = \{(\mathbf{Z}_{\text{org},1}, y_1), \cdots, (\mathbf{Z}_{\text{org},m}, y_m)\} = \{L_l\}_{l=1}^{c}$, where L_l denotes the set of texts labeled with l, y is the given label, and c is the number of categories. These texts with ground-truth labels in $\mathcal{D}_{lab}$ are called seed samples. Unlike the classic k-means algorithm, in the *centroids initialization* stage, instead of randomly selecting k samples from the data as the initial centroids, we use seed samples with the same label to form the initial clusters, and then average the vectors within the clusters as initial centroids. Moreover, in the *cluster updating* stage, we only update the cluster assignments of non-seed samples, keeping the cluster assignments of seed samples fixed in all iterations. The rest steps of the process remain consistent with the standard k-means algorithm. After completing the above process, we can assign the same weak label to data belonging to the same cluster for cluster-oriented CL.

We employ another projection head $\Psi(\cdot)$ to map the original embeddings into the hidden space, and normalize these hidden representations, *i.e.*, $\mathbf{Q} = \text{norm}(\Psi(\mathbf{Z}_{\text{org}}))$. The objective function of cluster-oriented CL can be expressed mathematically as:

$$\mathcal{L}_{ccl} = -\sum_{i}^{N} \frac{1}{|S_i| - 1} \sum_{j \in S_i} \log \frac{\exp(\mathbf{Q}_i \cdot \mathbf{Q}_j/\tau)}{\sum_{k=1}^{N} \mathbb{I}_{k \neq i} \exp(\mathbf{Q}_i \cdot \mathbf{Q}_k/\tau)} \tag{7}$$

where S_i symbols the set with the same label as sample i, but excluding sample i.

Model Training

For the original labeled texts, we also introduce a projection head $\Upsilon(\cdot)$ to map the learned representations into the hidden space where the cross-entropy loss function is applied, which can be formulated as:

$$\begin{aligned} \mathbf{R} &= \sigma(\mathbf{W}_{ce} \Upsilon(\mathbf{Z}_{\text{org}})) \\ \mathcal{L}_{ce} &= -\sum_{i \in \mathcal{D}_{lab}} \sum_{j}^{c} \mathcal{Y}_{ij} \log \mathbf{R}_{ij} \end{aligned} \tag{8}$$

Algorithm 1: The Training of GIFT

Input: The corpus $\mathcal{D} = \{d_i\}_{i=1}^{N}$
Output: The well-trained model

while *not done* **do**
  for $\pi \in \{w, e, p\}$ **do**
    Build the component graph $\mathcal{G}_\pi = \{\mathcal{V}_\pi, \mathbf{X}_\pi, \mathbf{A}_\pi\}$
  end for
  Update node embeddings for each component graph using Eq. 1.
  Construct TD matrices concerning the text and node
  Obtain Text representations using Eq. 2.
  Perform SVD for TD matrices using Eq. 4.
  Obtain the augmented views of texts using Eq. 5.
  Conduct CL using Eq. 6.
  Perform constrained seed k-means for the corpus.
  Assign weak labels for unlabeled texts.
  Conduct cluster-oriented CL using Eq. 7.
  Conduct the cross-entropy loss using Eq. 8.
  Optimize the model by the loss of Eq. 9.
end while
return: The well-trained GIFT.

where $\mathcal{Y}$ is the ground-truth label and $\mathbf{W}_{ce}$ is the trainable parameter.

The model is optimized by the combination of three loss functions, which can be denoted as:

$$\mathcal{L} = \eta\mathcal{L}_{cl} + \zeta\mathcal{L}_{ccl} + \mathcal{L}_{ce} \tag{9}$$

where η and ζ are control parameters.

When evaluating the model performance, we input the derived text embeddings of the test set into the classifier $\Upsilon(\cdot)$ to obtain the corresponding metrics. The pseudo-code of the training process of our model is presented in Algorithm 1.

Experiments

Datasets. To verify the effectiveness of our proposed model, we conduct experiments on several benchmark datasets, which are widely used in STC tasks. The statistics of these datasets are summarized in Table 1 and described in detail below. (1) **Twitter** is a binary classification dataset comprised of numerous tweets expressing two sentiments collected by the NLTK. (2) **MR** (Pang and Lee 2005) is a binary classification dataset of movie reviews, where each review contains a sentence that is labeled as positive or negative. (3) **Snippets** (Phan, Nguyen, and Horiguchi 2008) consists of web search snippets returned by the Google search engine. (4) **StackOverflow** (Xu et al. 2017) contains twenty categories of question titles crawled from the StackOverflow website.

Following previous studies (Wang et al. 2021), we randomly select 40 labeled data for each category of the dataset, half of which are used for training, another half for validation, and the remaining data are used for testing, to simulate the real situation with few labeled samples.

Baselines. We compare the proposed GIFT with the following types of models to demonstrate its superiority. (I) *Traditional models*: **TF-IDF+SVM** and **LDA+SVM** respectively utilize TF-IDF features and LDA features to represent text, and then train an SVM (Cortes and Vapnik 1995) for classification. **PTE** (Tang, Qu, and Mei 2015) learns word embeddings on a heterogeneous text graph, and averages the word embeddings as document embeddings. (II) *Deep learning models*: **CNNs** (Kim 2014) and **LSTM** (Liu et al. 2015) initialize texts using pre-trained GloVe word embeddings, then feed them into the corresponding deep networks. **BERT** (Devlin et al. 2019) and **RoBERTa** (Liu et al. 2019) are pre-trained on large corpus and can generate contextual embeddings when applied to specific tasks. Here, we use BERT-base and RoBERTa-base, which are fine-tuned along with the classifier in short texts. (III) *Graph-based models* contain **TLGNN** (Huang et al. 2019), **HyperGAT** (Ding et al. 2020), **TextING** (Zhang et al. 2020), **DADGNN** (Liu et al. 2021), and **TextGCN** (Yao, Mao, and Luo 2019). (IV) *Deep short text models* includes **STCKA** (Chen et al. 2019), **STGCN** (Ye et al. 2020), **HGAT** (Hu et al. 2019), **SHINE** (Wang et al. 2021), and **NC-HGAT** (Su et al. 2022).

Implementation Details. We adopt two-layer GCNs to encode each component graph, where the hidden dimension is set to 128. The temperature τ in CL and cluster-oriented CL are uniformly set to 0.5. All projection heads are implemented by an MLP with a hidden layer. In practice, performing SVD on large TD matrices requires particularly large time complexity. Therefore, an alternative way is to use randomized SVD (Halko, Martinsson, and Tropp 2011), which identifies a subspace that captures the input matrix dominant features by random sampling. It then projects the matrix into that subspace and performs SVD. In our case, the required rank of the approximate matrix is set to 15. The control parameter of loss function η, ζ are both set to 0.5. We use the Adam method to optimize GIFT with the learning rate 0.001. For other baselines, we use default parameters or conduct grid search for best-performing parameters. The evaluation metrics are accuracy (ACC) and macro F1-score (F1), widely adopted by previous studies.

Results

Model Performance. We conduct extensive experiments on several benchmark datasets to compare our proposed model with other baselines. All experiments are repeated ten times to obtain average metrics, which are shown in Table 2. Based on these quantitative observations, we have the following analyses.

Our model yields the best performance across all evaluation datasets in terms of the relevant metrics, demonstrating its superiority for the STC task. The success of our model can be attributed to several factors. First, we perform SVD on the text-specific TD matrices to obtain the augmented view of the text, which not only preserves useful information in the matrix but also removes noisy information. Second, we also introduce global TD co-occurrence signals to rectify the learning process of CL that may have been misguided in previous methods. Moreover, we explicitly leverage the prior knowledge contained in few labeled texts to assign weak labels to numerous unlabeled texts. This allows the

Dataset	Twitter	MR	Snippets	StackOverflow
#Doc	10,000	10,662	12,340	20,000
#Train(ratio)	40 (0.40%)	40 (0.38%)	160 (1.30%)	400 (2%)
#Word	21,065	18,764	29,040	2,632
#Entity	5,837	6,415	9,737	3,229
#POS Tag	41	41	34	42
Avg.Length	3.5	7.6	14.5	8.3
#Class	2	2	8	20

Table 1: Statistics of evaluation datasets.

Model	Twitter		MR		Snippets		StackOverflow	
	ACC	F1	ACC	F1	ACC	F1	ACC	F1
TF-IDF+SVM	53.62	52.46	54.29	48.13	64.70	59.17	59.19	59.06
LDA+SVM	54.34	53.97	54.40	48.39	62.54	56.40	60.19	59.52
PTE	54.24	53.17	55.02	52.62	63.10	59.11	62.56	61.32
CNN	57.29	56.02	59.06	59.01	77.09	69.28	63.75	61.21
LSTM	60.28	60.22	60.89	60.70	75.89	67.72	61.62	60.49
BERT	54.92	51.16	51.69	50.65	79.31	78.47	66.94	67.26
RoBERTa	56.02	52.29	52.55	51.30	79.55	79.02	69.91	70.35
TLGNN	59.02	54.56	59.22	59.36	70.25	63.29	62.09	61.91
HyperGAT	59.15	55.19	58.65	58.62	70.89	63.42	63.25	62.10
TextING	59.62	59.22	58.89	58.76	71.10	70.65	65.37	64.63
DADGNN	59.51	55.32	58.92	58.86	71.65	70.66	66.26	65.10
TextGCN	60.15	59.82	59.12	58.98	77.82	71.95	67.02	66.51
STCKA	57.56	57.02	53.25	51.19	68.96	61.27	59.72	59.65
STGCN	64.33	64.29	58.25	58.22	70.01	69.93	69.23	69.10
HGAT	63.21	57.02	62.75	62.36	82.36	74.44	67.35	66.92
SHINE	72.54	72.19	64.58	63.89	82.39	81.62	73.05	72.73
NC-HGAT	63.76	62.94	62.46	62.14	82.42	74.62	67.59	67.02
GIFT	**73.16**	**73.16**	**65.21**	**65.16**	**83.73**	**82.35**	**83.07**	**82.94**

Table 2: Results (%) of the Accuracy and Macro-F1 score on several short text datasets. We highlight the best performance in bold based on the pairwise t-test with 95% confidence.

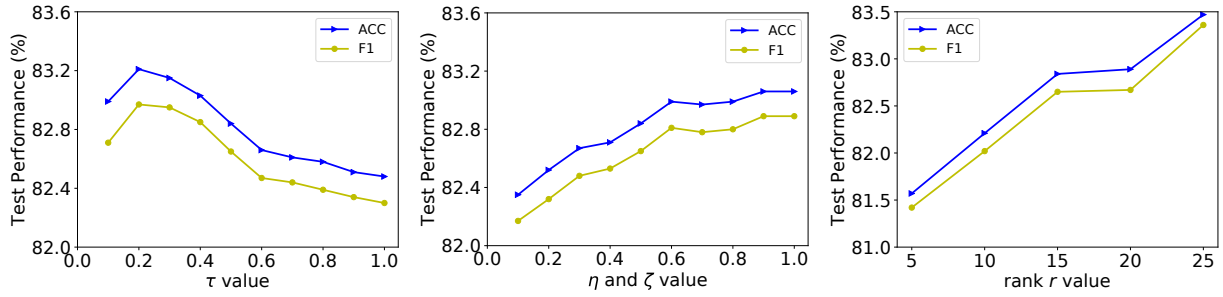
model to utilize clustering-friendly features through our designed cluster-oriented CL. By doing so, we alleviate the issue of false negatives caused by the instance-discrimination CL paradigm. Furthermore, we construct a heterogeneous graph consisting of several component graphs to fully exploit the semantic and syntactic information from the text itself and external knowledge graphs to correctly identify the meaning of the text.

We find that deep short text models, especially those combined with GNNs like HGAT and NC-HGAT, achieve competitive performance compared to other categories of models. A reasonable explanation is that they are specially designed for STC tasks and enrich short text representations by introducing various auxiliary knowledge. Notably, NC-HGAT, which performs CL with random perturbations on the built corpus-level graph, lags far behind our model, demonstrating that its augmented view may discard texts' important information, hindering model learning. Additionally, some fine-tuned pre-trained models, such as BERT and RoBERTa, which incorporate generic knowledge from a large corpus, perform unfavorably on the focused task due to the lack of labeled texts. Moreover, graph-based models are competitive because they explicitly model syntactic structure information and can benefit from label propagation.

Ablation Study. To verify the effectiveness of each designed individual module of our model, we conduct the following ablation experiments on several model variants. (I) *GIFT-SVD-K*: We simultaneously remove both CL with augmented views generated by SVD and cluster-oriented CL with constrained seed k-means, and instead directly classify the obtained text representations using cross-entropy loss. (II) *GIFT-K*: We only eliminate cluster-oriented CL while keeping other parts unchanged. (III) *GIFT-SVD*: We only exclude CL performed with SVD while preserving the remaining parts. (IV) $GIFT_{random}$: We replace SVD with random perturbations, *i.e.*, randomly masking the TD matrix, to generate augmented views. (V) $GIFT_{k\text{-}means}$: We replace the constrained seed k-means in cluster-oriented CL with classic k-means. According to the results shown in Table 3, we can draw the following findings. First, removing any component leads to performance degradation, demonstrating that they all play essential roles in GIFT. Second, we can find that GIFT performs better than $\text{GIFT}_{\text{k-means}}$ and $\text{GIFT}_{\text{random}}$, which is consistent with our expectations and validate our previous arguments.

Model	Twitter		MR		Snippets		StackOverflow	
	ACC	F1	ACC	F1	ACC	F1	ACC	F1
GIFT-SVD-K	69.60	69.47	61.52	61.48	79.25	78.62	76.20	75.32
GIFT-K	71.49	71.40	63.24	63.19	82.89	81.25	81.42	81.20
GIFT-SVD	71.95	71.90	64.02	63.96	82.61	81.26	81.72	81.52
$\text{GIFT}_{\text{random}}$	72.11	72.09	64.32	64.19	82.85	81.75	82.02	81.90
$\text{GIFT}_{\text{k-means}}$	72.76	72.62	64.62	64.59	83.28	81.92	82.52	82.30
GIFT	**73.16**	**73.16**	**65.21**	**65.16**	**83.73**	**82.35**	**83.07**	**82.94**

Table 3: The ablation results (%) of various experimental settings.



(a) Temperature parameter (b) Control coefficient (c) Rank Value

Figure 2: Hyperparameter Sensitivity on StackOverflow dataset.

Hyperparameter Study. We investigate the impact of several key parameters, *i.e.*, the temperature τ, the control coefficients η, ζ, and the rank value r, on the model performance. From Fig. 2 (a), we observe that with the increase of temperature τ, the model performance first increases and then decreases. A plausible reason is that small temperatures make the model focus on hard negative samples while pushing away potential positive samples with shared semantics, and large temperatures make the model treat negative samples equally and reduce its distinguishable ability. Fig. 2 (b) and (c) both show a trend of results increasing with hyperparameters increasing. The former is because as η and ζ increase, the contribution of the two kinds of CL to the final loss function increases accordingly; the latter is because as the rank r increases, the augmented TD matrix can capture more global co-occurrence signals.

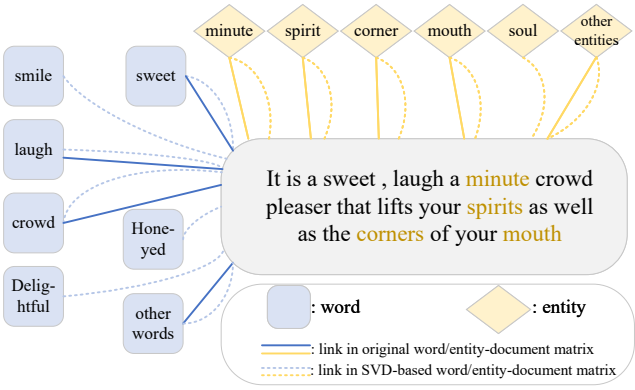


Figure 3: Case study of a short text from the MR dataset.

Case Study. We conduct a case study to intuitively demonstrate the effectiveness of our model in injecting global co-occurrence information. We present two built TD matrices (*i.e., word-document and entity-document matrix*) for clarity. As shown in Fig. 3, we can clearly observe that in the SVD-reconstructed word-document matrix, GIFT re-establishes the connections between potentially important words and documents, such as "delightful", "honeyed" and "smile", the first two enhancing the "sweet" in the text, and the last one enhancing "laugh". In addition, in the rebuilt entity-document matrix, our model also re-establishes the connection between "soul" and documents. These global co-occurrence signals play a crucial role in accurately comprehending short texts.

Conclusion

In this work, we propose a novel model GIFT for STC tasks. Our model leverages a unique approach to obtain augmented views of texts by applying SVD on text-specific TD matrices for CL. Meanwhile, we incorporate cluster-oriented CL based on constrained k-means to explore clustering-friendly features in the data. Extensive experiments demonstrate that the proposed model significantly outperforms other state-of-the-art models.

Acknowledgments

This work is supported in part by funds from the National Key Research and Development Program of China (No. 2021YFF1201200), the National Natural Science Foundation of China (No. 62172187 and No. 62372209). Fausto Giunchiglia's work is funded by European Union's Horizon 2020 FET Proactive project (No. 823783).

References

Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; and Yakhnenko, O. 2013. Translating embeddings for modeling multi-relational data. In *NeurIPS*.

Cai, X.; Huang, C.; Xia, L.; and Ren, X. 2023. LightGCL: Simple Yet Effective Graph Contrastive Learning for Recommendation. In *ICLR*.

Caron, M.; Misra, I.; Mairal, J.; Goyal, P.; Bojanowski, P.; and Joulin, A. 2020. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*.

Chen, J.; Hu, Y.; Liu, J.; Xiao, Y.; and Jiang, H. 2019. Deep short text classification with knowledge powered attention. In *AAAI*.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. In *ICML*.

Chen, X.; and He, K. 2021. Exploring simple siamese representation learning. In *CVPR*.

Cortes, C.; and Vapnik, V. 1995. Support-vector networks. *Machine Learning*, 20: 273–297.

Deerwester, S.; Dumais, S. T.; Furnas, G. W.; Landauer, T. K.; and Harshman, R. 1990. Indexing by latent semantic analysis. *JASIS*, 41(6): 391–407.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Ding, K.; Wang, J.; Li, J.; Li, D.; and Liu, H. 2020. Be more with less: Hypergraph attention networks for inductive text classification. In *EMNLP*.

Entezari, N.; Al-Sayouri, S. A.; Darvishzadeh, A.; and Papalexakis, E. E. 2020. All you need is low (rank) defending against adversarial attacks on graphs. In *WSDM*.

Gao, T.; Yao, X.; and Chen, D. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *EMNLP*.

Gunel, B.; Du, J.; Conneau, A.; and Stoyanov, V. 2021. Supervised contrastive learning for pre-trained language model fine-tuning. In *ICLR*.

Halko, N.; Martinsson, P.-G.; and Tropp, J. A. 2011. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2): 217–288.

Hansen, P. C. 1987. The truncated SVD as a method for regularization. *BIT Numerical Mathematics*, 27: 534–553.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *CVPR*.

Hu, L.; Yang, T.; Shi, C.; Ji, H.; and Li, X. 2019. Heterogeneous graph attention networks for semi-supervised short text classification. In *EMNLP-IJCNLP*.

Huang, L.; Ma, D.; Li, S.; Zhang, X.; and Wang, H. 2019. Text Level Graph Neural Network for Text Classification. In *EMNLP-IJCNLP*.

Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; and Krishnan, D. 2020. Supervised contrastive learning. In *NeurIPS*.

Kim, Y. 2014. Convolutional Neural Networks for Sentence Classification. In *EMNLP*.

Li, J.; Zhou, P.; Xiong, C.; and Hoi, S. 2021. Prototypical Contrastive Learning of Unsupervised Representations. In *ICLR*.

Liu, P.; Qiu, X.; Chen, X.; Wu, S.; and Huang, X.-J. 2015. Multi-timescale long short-term memory neural network for modelling sentences and documents. In *EMNLP*.

Liu, Y.; Di Liang, M. L.; Giunchiglia, F.; Li, X.; Wang, S.; Wu, W.; Huang, L.; Feng, X.; and Guan, R. 2023a. Local and Global: Temporal Question Answering via Information Fusion. In *IJCAI*.

Liu, Y.; Guan, R.; Giunchiglia, F.; Liang, Y.; and Feng, X. 2021. Deep attention diffusion graph neural networks for text classification. In *EMNLP*.

Liu, Y.; Liang, D.; Fang, F.; Wang, S.; Wu, W.; and Jiang, R. 2023b. Time-aware multiway adaptive fusion network for temporal knowledge graph question answering. In *ICASSP*.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Minaee, S.; Kalchbrenner, N.; Cambria, E.; Nikzad, N.; Chenaghlu, M.; and Gao, J. 2021. Deep learning–based text classification: a comprehensive review. *ACM CSUR*, 54(3): 1–40.

Pang, B.; and Lee, L. 2005. Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales. In *ACL*.

Phan, X.-H.; Nguyen, L.-M.; and Horiguchi, S. 2008. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *The Web Conference*.

Su, Z.; Harit, A.; Cristea, A. I.; Yu, J.; Shi, L.; and Al Moubayed, N. 2022. Contrastive learning with heterogeneous graph attention networks on short text classification. In *IJCNN*.

Tang, J.; Qu, M.; and Mei, Q. 2015. Pte: Predictive text embedding through large-scale heterogeneous text networks. In *SIGKDD*.

Wang, J.; Wang, Z.; Zhang, D.; and Yan, J. 2017. Combining Knowledge with Deep Convolutional Neural Networks for Short Text Classification. In *IJCAI*.

Wang, Y.; Wang, S.; Yao, Q.; and Dou, D. 2021. Hierarchical Heterogeneous Graph Representation Learning for Short Text Classification. In *EMNLP*.

Wu, Z.; Shu, L.; Xu, Z.; Chang, Y.; Chen, C.; and Zheng, Z. 2022. Robust tensor graph convolutional networks via t-svd based graph augmentation. In *SIGKDD*.

Xu, J.; Xu, B.; Wang, P.; Zheng, S.; Tian, G.; and Zhao, J. 2017. Self-taught convolutional neural networks for short text clustering. *Neural Networks*, 88: 22–31.

Yao, L.; Mao, C.; and Luo, Y. 2019. Graph convolutional networks for text classification. In *AAAI*.

Ye, Z.; Jiang, G.; Liu, Y.; Li, Z.; and Yuan, J. 2020. Document and word representations generated by graph convolutional network and bert for short text classification. In *ECAI*.

Yu, J.; Yin, H.; Xia, X.; Chen, T.; Cui, L.; and Nguyen, Q. V. H. 2022. Are graph augmentations necessary? simple graph contrastive learning for recommendation. In *SIGIR*.

Zeng, J.; Li, J.; Song, Y.; Gao, C.; Lyu, M. R.; and King, I. 2018. Topic Memory Networks for Short Text Classification. In *EMNLP*.

Zhang, D.; Nan, F.; Wei, X.; Li, S.-W.; Zhu, H.; Mckeown, K.; Nallapati, R.; Arnold, A. O.; and Xiang, B. 2021. Supporting Clustering with Contrastive Learning. In *NAACL*.

Zhang, Y.; Yu, X.; Cui, Z.; Wu, S.; Wen, Z.; and Wang, L. 2020. Every Document Owns Its Structure: Inductive Text Classification via Graph Neural Networks. In *ACL*.