Boosting Short Text Classification with Multi-Source Information Exploration and Dual-Level Contrastive Learning

Yonghao Liu[1*], Mengyu Li[1*], Wei Pang[2], Fausto Giunchiglia[3], Lan Huang[1], 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]Mathematical and Computer Sciences, Heriot-Watt University
[3]University of Trento
{yonghao20, mengyul21}@mails.jlu.edu.cn, w.pang@hw.ac.uk, fausto.giunchiglia@unitn.it
{huanglan, fengxy, guanrenchu}@jlu.edu.cn

Abstract

Short text classification, as a research subtopic in natural language processing, is more challenging due to its semantic sparsity and insufficient labeled samples in practical scenarios. We propose a novel model named MI-DELIGHT for short text classification in this work. Specifically, it first performs multi-source information (*i.e., statistical information, linguistic information, and factual information*) exploration to alleviate the sparsity issues. Then, the graph learning approach is adopted to learn the representation of short texts, which are presented in graph forms. Moreover, we introduce a dual-level (*i.e., instance-level and cluster-level*) contrastive learning auxiliary task to effectively capture different-grained contrastive information within massive unlabeled data. Meanwhile, previous models merely perform the main task and auxiliary tasks in parallel, without considering the relationship among tasks. Therefore, we introduce a hierarchical architecture to explicitly model the correlations between tasks. We conduct extensive experiments across various benchmark datasets, demonstrating that MI-DELIGHT significantly surpasses previous competitive models. It even outperforms popular large language models on several datasets.

Introduction

Text classification is a fundamental task in natural language processing (NLP). As a special form of text, short texts often appear in our daily life in the form of tweets, queries, and news feeds (Phan, Nguyen, and Horiguchi 2008). To deal with these short texts, short text classification (STC), as a subtask of text classification, has attracted extensive attention from the research community. It is widely used in various practical scenarios, such as news classification (Dilrukshi, De Zoysa, and Caldera 2013), sentiment analysis (Chen et al. 2019), and query intent classification (Wang et al. 2017). It is worth noting that compared to traditional text classification, STC is particularly nontrivial, which is mainly attributed to its two well-known challenges, *i.e.*, *semantic sparsity* and *limited labeled texts* (Hu et al. 2019).

For the challenge of *semantic sparsity*, short texts typically contain only one or two sentences with few words, which have limited available contextual information (Tang, Qu, and Mei 2015). Such severe semantic sparsity often leads to vagueness and ambiguity, thus hindering the accurate understanding of short texts. An effective solution is to explore multi-source information to enrich the context for short texts. On the one hand, we can collect the *statistical information* and *linguistic information* contained within short texts. *Statistical information* is related to the statistics of words that constitute texts, and is often obtained by modeling word co-occurrence and word distribution probabilities in the text (Thilakaratne, Falkner, and Atapattu 2019; Liu et al. 2024d). For example, by analyzing the word statistics in the text, Latent Dirichlet Allocation (LDA) can uncover the topic structure that can enrich the information in short texts. *Linguistic information* is implicit in the syntax and semantics of texts (Liu et al. 2019). For instance, we can obtain the part-of-speech (POS) information of words to determine their syntactic roles in the text. On the other hand, auxiliary *factual information* can also be injected to compensate for the missing contextual information (Liu et al. 2023b,a). In this paper, factual information mainly refers to those text-related entities existing in common sense knowledge graphs (Chen et al. 2019). With such enriched auxiliary information, the learned model can naturally understand the meanings of short texts better.

When faced with the challenge of limited labeled texts in real-world applications, it is often the case that there is a vast amount of easily accessible short texts, but only a small number of labeled data are available (Kenter and De Rijke 2015). In addition, the proportion of unlabeled data is much higher compared to that of long texts. Consequently, deep learning models that rely on large-scale labeled data for training are susceptible to overfitting issues, leading to unsatisfactory performance outcomes. To cope with such issue, *on the one hand*, some works (Hu et al. 2019; Yang et al. 2021) are mainly devoted to fully utilizing limited labeled short texts. They perform supervised graph learning on the constructed corpus-level graph to learn the textual representations. Nevertheless, the performance of these models is

[*]These authors contributed equally.
[†]Corresponding Author.

largely influenced by the limited labeled data, as they only provide relatively restricted information. *On the other hand*, some works (Liu, Qiu, and Huang 2016, 2017) attempt to introduce auxiliary tasks to alleviate the inefficient data problem. They typically design auxiliary tasks and then jointly train these tasks, aiming to enable the knowledge contained in the tasks to be utilized by other tasks, thereby improving the model generalization ability. However, the reliability of auxiliary tasks are questionable, and unreliable auxiliary ones can impair the model performance.

Recently, contrastive learning (CL) has attracted tremendous attention due to its effectiveness in extracting features from unlabeled data and simple mechanism. Using CL for auxiliary feature learning appears to be a promising approach, as it enables the extraction of self-supervised contrastive information from a large corpus of unlabeled texts. Moreover, CL has been extensively demonstrated to function as a dependable auxiliary task for extracting discriminative information (Chen et al. 2022; Pan et al. 2022). By this way, we can simultaneously handle the limitations of the two aforementioned types of models. However, typically, only instance-level contrastive learning (ICL) is previously used for auxiliary feature learning, which regards each instance as a distinct class. The unique positive pair originates from the same instance, and other instances sharing similar underlying semantics are considered negative pairs and pushed apart. Therefore, it is not sufficient to use such an unsupervised ICL approach from a fine-grained perspective alone. We also need to introduce a coarse-grained CL as an auxiliary task, such as cluster-level supervised CL (CCL), which can further enable the aggregation of samples that share intrinsically similar signals from a coarse-grained perspective. Furthermore, previous models that incorporate CL simply combine the losses of the main task and the auxiliary task, performing them in parallel, without adequately considering the significance of a well-structured architecture that facilitates connections among multiple tasks. This approach is deemed unreasonable, as there exists a causal relationship established through the learned features across tasks. In other words, as we progress from ICL to CCL to classification, the growing complexity of the tasks necessitates the acquisition of increasingly sophisticated features, enabling a transition from rudimentary to abstract characteristics.

In this paper, we introduce a novel model called **MI-DELIGHT** that leverages **M**ulti-source **I**nformation and **D**ual-level contrastiv**E** **L**earn**I**ng for s**H**ort **T**ext classification. On the one hand, graphs, as a basic data structure, possess the desirable characteristics of flexibility and simplicity. As such, we adopt graphs as the uniform representation form for texts with injected information. On the other hand, graph neural networks (GNNs) have a natural advantage in learning from graph data. Meanwhile, in numerous NLP tasks, GNNs have demonstrated superior performance in capturing non-consecutive and long-range word interactions, as well as their powerful representation capabilities for modeling texts. Therefore, we first construct a word graph and a POS graph to explore the statistical and linguistic information contained in short texts. Additionally, we also build an entity graph to introduce supplementary factual information. After obtaining all the information mentioned above by GNNs and extracting rudimentary text features, we design a dual-level CL auxiliary task to assist in obtaining improved text features in a more comprehensive manner. Importantly, we introduce a hierarchical structure to leverage the casual relationships among tasks and extract abstract features step by step. Specifically, we first employ ICL based on the elementary text features to capture the fine-grained contrastive information. Then, we perform CCL based on advanced text features obtained during the ICL process to capture the coarse-grained contrastive information. Finally, we classify high-level text features obtained during the CCL process. In summary, our contributions are as follows:

(1) We propose a novel model, namely MI-DELIGHT, which is capable of modeling short texts and resolving existing semantic sparsity and inefficient labeled samples.

(2) We build three types of graphs to explore statistical, linguistic and factual information to compensate for critical context. Moreover, we design a hierarchical dual-level CL auxiliary tasks, including CCL and ICL, to effectively capture multi-grained contrastive information.

(3) We conduct diverse experiments, and MI-DELIGHT consistently surpasses other competitive models, including some popular large language models, across several benchmark datasets.

Related Work

Text Classification: Traditional text classification methods typically first use hand-crafted lexical features (Li et al. 2022), such as BoW and TF-IDF, to represent text and then adopt SVM or Naive Bayes classifiers. With the development of neural networks, deep learning models without feature engineering, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have become mainstream approaches in this area. Moreover, recent studies (Guan et al. 2021; Liu et al. 2022) have demonstrated the successful application of GNNs to text classification tasks. These models capture word interactions using graph structures and have shown promising results. One line of work involves building corpus graphs, treating both text and words as nodes, and performing classification in a semi-supervised manner (Yao, Mao, and Luo 2019; Liu et al. 2020). Another line of research focuses on constructing a graph for each text and deriving document representations through graph learning on word-word edges (Ding et al. 2020; Liu et al. 2021).

Short Text Classification: STC is a challenging task in which irregular word orders and missing function words hinder the proper understanding of short texts. Some existing and popular approaches attempt to introduce additional information, such as entities or latent topics, to assist with text understanding (Zeng et al. 2018). Some studies (Ye et al. 2020; Wang et al. 2021) have conducted label propagation via graph structures of constructed heterogeneous graphs and yielded notable gains. Further, several models (Su et al. 2022; Liu et al. 2024b) propose leveraging CL on the corpus-level graph for STC and achieves promising results. However, these models only engage in instance-level CL, thus disregarding cluster-level features. Recent popular

large language models (LLMs), such as GPT-3.5 (Ouyang et al. 2022) and Llama (Touvron et al. 2023) have been pretrained on massive high-quality data, thus exhibiting excellent understanding of general texts. However, their performance on domain-specific (*e.g*, medical or legal domains) texts is not as expected (Chang et al. 2023).

Contrastive Learning: CL approaches learn representations by contrasting positive pairs against negative pairs, and have been highly successful in various fields such as NLP (Gao, Yao, and Chen 2021; Wu et al. 2022) and graphs (Liu et al. 2024c,a). Initially, many CL approaches focus on instance discrimination tasks in an unsupervised manner (Caron et al. 2020; Tian, Krishnan, and Isola 2020). The following studies (Khosla et al. 2020; Liu et al. 2024c; Li et al. 2024) explore fully supervised CL, which can explicitly leverage label information, enabling the extraction of more task-relevant information. Some recent studies (Zheng et al. 2021; Huynh et al. 2022) consider similar samples as positive pairs and aim to pull them together. However, these methods mostly focus on unsupervised tasks and do not take advantage of the handful of available instance labels.

Preliminary

Most modern GNN models follow a recursive neighborhood aggregation scheme, where the representation of a node is iteratively updated by aggregating the features of its neighbors. A classic model is the graph convolutional network (GCN) (Kipf and Welling 2017), which can be defined formally 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 $\mathbf{H}^{(\ell)}$ is the ℓ-th output node representation and $\mathbf{H}^{(0)} = \mathbf{X}$ is an initial node embedding. $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ is an adjacency matrix with added self-loops, and $\hat{\mathbf{D}}_{ii} = \sum_j \hat{\mathbf{A}}_{ij}$ is the corresponding diagonal degree matrix. $\sigma(\cdot)$ is an activation function such as ReLU and $\mathbf{W}^{(\ell)}$ is a layer-specific trainable matrix.

Method

In this section, we present the proposed MI-DELIGHT model for STC. The overall architecture is shown in Fig. 1. We then proceed to elaborate on the key components.

Multi-Source Information Exploration

Our goal is to develop a model that can efficiently predict the labels of numerous unlabeled texts when trained on a given short text dataset $\mathcal{D}$ with limited labeled samples. To alleviate the issue of semantic sparsity in short texts, we aim to perform multi-source information exploration to maximally utilize statistical and linguistic information from the text itself, as well as factual information from outside.

Statistical Information: As mentioned before, statistical information is related to word statistics in the text. To capture this information, we construct a word graph $\mathcal{G}_w = \{\mathcal{V}_w, \mathbf{X}_w, \mathbf{A}_w\}$, where $\mathcal{V}_w$ is the set of word nodes and $\mathbf{A}_w \in \mathbb{R}^{|\mathcal{V}_w|\times|\mathcal{V}_w|}$ is the corresponding adjacency matrix determined by point-wise mutual information (PMI), *i.e.*, $\mathbf{A}_{w,ij} = \max(\text{PMI}(v_i, v_j), 0)$, where $v_i, v_j \in \mathcal{V}_w$, which is a popular way to measure the word co-occurrence relationship. $\mathbf{X}_w \in \mathbb{R}^{|\mathcal{V}_w|\times f_w}$ is the feature matrix of all words with f_w-dimensional features. We initialize $\mathbf{X}_w$ as pretrained word embeddings generated by the GloVe method, which explicitly utilizes the global co-occurrence information of words and captures the underlying statistical information. Then, we feed the word graph $\mathcal{G}_w$ into the GCN using Eq.1 to obtain updated node embeddings $\mathbf{H}_w$ with statistical information.

Linguistic Information: This information is necessary for comprehensively understanding short texts, including semantic and syntactic structure. Here, we acquire linguistic information by identifying the syntactic roles of each word in a given text, such as adjectives or adverbs, which helps to eliminate syntactic word ambiguity. To this end, we construct a POS graph $\mathcal{G}_p = \{\mathcal{V}_p, \mathbf{X}_p, \mathbf{A}_p\}$, where $\mathcal{V}_p$ is the formed POS tag node set and $\mathbf{A}_p$ denotes the POS adjacency matrix calculated by PMI, *i.e.*, $\mathbf{A}_{p,ij} = \max(\text{PMI}(v_i, v_j), 0)$, where $v_i, v_j \in \mathcal{V}_p$. We initialize the node features $\mathbf{X}_t \in \mathbb{R}^{|\mathcal{V}_t|\times f_t}$ as one-hot vectors. Similarly, we obtain updated POS tag features $\mathbf{H}_p$ by performing Eq.1.

Factual Information: Additional factual information can help supplement the contextual knowledge required for short texts to enhance the classification ability of subsequent models. Therefore, we extract the entities in the short text that are resided in the knowledge graph and construct an entity graph $\mathcal{G}_e = \{\mathcal{V}_e, \mathbf{X}_e, \mathbf{A}_e\}$. Here, we utilize the TAGME tool for entity linking on the NELL (Carlson et al. 2010) knowledge graph. $\mathcal{V}_e$ is the entity node set. The entities' embeddings $\mathbf{X}_e \in \mathbb{R}^{|\mathcal{V}_e|\times f_e}$ are initialized by TransE (Bordes et al. 2013). $\mathbf{A}_e$ is the entity adjacency matrix derived by the cosine similarity of each entity pair, *i.e.*, $\mathbf{A}_{e,ij} = \max(\cos(\mathbf{X}_{e,i}, \mathbf{X}_{e,j}), 0)$. The updated entity node embeddings $\mathbf{H}_e$ are also obtained by performing Eq.1.

Text Representation Learning

Given three types of graphs $\mathcal{G} = \{\mathcal{G}_\pi\}, \pi \in \{w, e, p\}$, to obtain text embeddings, we employ the following information aggregation strategy:

$$\begin{aligned} \mathbf{Z}_\pi &= \mathbf{P}_\pi \mathbf{H}_\pi, \\ \mathbf{Z}_\pi &= \mathbf{Z}_\pi / ||\mathbf{Z}_\pi||_2, \pi \in \{w, e, p\}, \end{aligned} \tag{2}$$

where $\mathbf{H}_\pi$ denotes the updated node embeddings of $\mathcal{G}_\pi$ obtained via a 2-layer GCN. We set $\mathbf{P}_\pi \in \mathbb{R}^{N\times|\mathcal{V}_\pi|}$ as the TF-IDF value between each text and word or POS tag of the corpus when $\pi \in \{w, p\}$. N denotes the number of short texts. Moreover, when $\pi = e$, we make $\mathbf{P}_{e,ij} = 1$ if the i-th text contains the j-th entity and 0 otherwise. After the normalized text-relevant features $\mathbf{Z}_w$, $\mathbf{Z}_e$, and $\mathbf{Z}_p$ are derived, we concatenate them to obtain the text embeddings, *i.e.*, $\mathbf{Z} = \mathbf{Z}_w||\mathbf{Z}_e||\mathbf{Z}_p$.

Data Augmentation

A key step for applying CL to NLP is to construct positive sample pairs. A typical approach for generating positive samples is a data augmentation technique, such as back-translation (Edunov et al. 2018), random noise injection (Xie

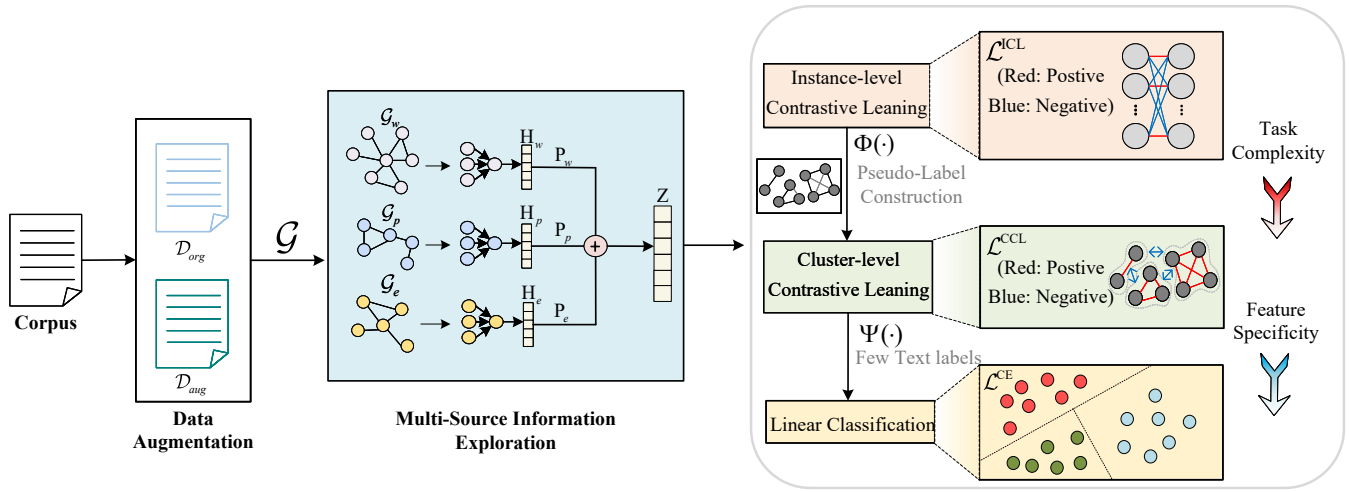

Figure 1: The overall architecture of MI-DELIGHT. We first generate augmented samples for the input texts. Then, the original corpus $\mathcal{D}_{\text{org}} = \{d_i^{\text{org}}\}_{i=1}^{N}$ and the augmented corpus $\mathcal{D}_{\text{aug}} = \{d_i^{\text{aug}}\}_{i=1}^{N}$ are used to construct a word graph $\mathcal{G}_w$, a POS graph $\mathcal{G}_p$ and an entity graph $\mathcal{G}_e$, and the text embeddings $\mathbf{Z}$ are obtained via the text representation learning module. Finally, these embeddings are mapped through different projection heads into different hidden spaces to which ICL, CCL, and cross-entropy (CE) are applied in a certain hierarchical order. From ICL to CCL and then to CE, the task complexity keeps increasing, and the features keep more abstract. Here, *feature specificity* represents the abstraction level of features.

et al. 2017), and word substitution (Wei and Zou 2019). Here, we augment the original data by replacing its words with WordNet synonyms. Formally, for each text d_i^{org} in the original corpus $\mathcal{D}_{\text{org}} = \{d_i^{\text{org}}\}_{i=1}^{N}$, we can obtain the augmented text $d_i^{\text{aug}} = \text{aug}\left(d_i^{\text{org}}\right)$ and augmented corpus $\mathcal{D}_{\text{aug}} = \{d_i^{\text{aug}}\}_{i=1}^{N}$. We denote the overall corpus and the corresponding text embeddings as $\mathcal{D} = \mathcal{D}_{\text{org}} \cup \mathcal{D}_{\text{aug}}$ and $\mathbf{Z} = \mathbf{Z}^{\text{org}} \cup \mathbf{Z}^{\text{aug}}$, respectively. Note that our model is feasible for data augmentations, and we explore the impacts of different data augmentations on the model in the experiment section.

Hierarchical Structure among Tasks

In contrast to prior models, we have implemented a *hierarchical structure* to explicitly account for the relationship established through distinct stages of learned text features among the primary classification task and the auxiliary CL tasks. First, we utilize the rudimentary features acquired during the multi-source information exploration stage to perform ICL, enabling us to capture fine-grained contrastive information. Then, based on the intermediate features obtained at the ICL stage, we perform CCL to capture coarse-grained contrastive information. Finally, leveraging abstract features obtained at the CCL stage, we carry out the ultimate classification task.

Instance-Level Contrastive Learning: First, based on the rudimentary text features $\mathbf{Z}$, we leverage ICL to perform instance discrimination tasks to explore fine-grained contrastive information. Typically, two texts from the same source data exhibit similar meanings. Their encoded text-level embeddings should be as similar as possible in the latent space. We refer to d_i and the augmented version d_j as a pair of positive samples while treating the other $2(N-1)$ texts in $\mathcal{D}$ as negative samples to this positive pair, which should be far away from the positive samples. With the obtained text embeddings $\mathbf{Z}$, we perform the normalization operation on them, *i.e.*, $\tilde{\mathbf{Z}} = \mathbf{Z}/||\mathbf{Z}||_2$. Notably, we do not map $\mathbf{Z}$ to a hidden space through a projection head as in traditional CL. Due to the fine-grained information required by ICL, introducing a projection head would not only compromise the semantics but also introduce more parameters. Therefore, we avoid using a projection head in this stage. The objective function for a positive pair of examples (d_i, d_j) is defined as follows:

$$\mathcal{L}_i^{\text{ICL}} = -\log \frac{\exp((\tilde{\mathbf{Z}}_i \cdot \tilde{\mathbf{Z}}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{I}_{k \neq i} \exp((\tilde{\mathbf{Z}}_i \cdot \tilde{\mathbf{Z}}_k)/\tau)}, \tag{3}$$

where $\tilde{\mathbf{Z}}_i$ and $\tilde{\mathbf{Z}}_j$ denote the output embeddings of the i-th text and its augmented text, respectively. $\mathbb{I}_{k \neq i}$ is an indicator function set to 1 if $k \neq i$, and τ denotes the temperature parameter.

The ICL loss is computed by averaging over all positive pairs on $\mathcal{D}$, which is expressed as:

$$\mathcal{L}^{\text{ICL}} = \frac{1}{2N} \sum_{i=1}^{2N} \mathcal{L}_i^{\text{ICL}}. \tag{4}$$

Cluster-Level Contrastive Learning: Next, based on the intermediate text features $\tilde{\mathbf{Z}}$ derived in the ICL stage corresponding to the corpus $\mathcal{D}$, we perform CCL. We expect to assign pseudo-cluster labels to the data in the corpus to explore their similarities from a cluster perspective and exploit their high-level feature clustering information such that similar instances can be pulled together. For ease of the presentation, we denote $\tilde{\mathbf{Z}}^*$ as the original or augmented text features from $\mathcal{D}_*$, where $*$ stands for "org" or "aug". We

leverage the scores computed by the cosine similarity function to build relations between different texts. Next, we define a text d_j^* as the nearest neighbor of text d_i^* when Eq.5 is satisfied.

$$\begin{aligned} \text{near}(d_i^*) &= \arg\max_{d_j^*} \cos(\tilde{\mathbf{Z}}_i^*, \tilde{\mathbf{Z}}_j^*) \\ &\textit{s.t.} \quad \forall d_j^* \in \mathcal{D}_* \wedge j \neq i. \end{aligned} \tag{5}$$

The text d_i^* is connected with the text d_j^* if $\text{near}(d_i^*) = d_j^*$ or $\text{near}(d_j^*) = d_i^*$. After performing the above operation, we can acquire the symmetric connections within $\mathcal{D}_*$. Then, we utilize the connected component labeling algorithm (Di Stefano and Bulgarelli 1999) to assign the corresponding pseudo-cluster label for each derived component. Since any two instances in a component can be connected by paths, we treat its internal instances as similar. Subsequently, we can obtain the label matrix $\mathbf{Y}^*$. $\mathbf{Y}_{ij}^*$ is set to 1 if d_i^* and d_j^* are in the same component.

Subsequently, we adopt a projection head $\Phi(\cdot)$, which maps the representations $\tilde{\mathbf{Z}}$ to a hidden space where the CCL loss is applied, *i.e.*, $\mathbf{U} = \Phi(\tilde{\mathbf{Z}})$. Due to the different target granularities, there may be potential conflicts in the feature space between CCL and ICL, thus requiring a projection head. Here, the dimension of $\mathbf{U}$ is half the dimension of $\tilde{\mathbf{Z}}$. Next, we normalize the output into a unit form, *i.e.*, $\tilde{\mathbf{U}} = \mathbf{U}/||\mathbf{U}||_2$. Moreover, intuitively, since d_i^{org} and d_i^{aug} have the same meanings, the labels $\mathbf{Y}_i^{\text{org}}$ and $\mathbf{Y}_i^{\text{aug}}$ should be consistent. We can swap supervision signals between them. Another crucial reason for swapping supervision is that the dot product value of the positive samples in the same class may be large, which leads to a potentially small $\mathcal{L}_i^{\text{CCL}}$ under standard CL settings, thus affecting model optimization. The detailed CCL loss with swapped supervision is defined as follows:

$$\begin{aligned} \mathcal{L}^{\text{CCL}} &= \frac{1}{N}\sum\nolimits_{i=1}^{N} \mathcal{L}_i^{\text{CCL}}, \\ \mathcal{L}_i^{\text{CCL}} &= \mathcal{L}_i^{\text{swap}}(\mathbf{U}_{\text{org},i}, \mathbf{Y}_i^{\text{aug}}) + \mathcal{L}_i^{\text{swap}}(\mathbf{U}_{\text{aug},i}, \mathbf{Y}_i^{\text{org}}), \\ \mathcal{L}_i^{\text{swap}} &= -\sum_{j}^{N} \mathbb{I}_{\mathbf{Y}_{ij}=1} \log \frac{\exp(\mathbf{U}_i \cdot \mathbf{U}_j/\tau)}{\sum_{k=1}^{N} \mathbb{I}_{k\neq i} \exp(\mathbf{U}_i \cdot \mathbf{U}_k/\tau)}, \end{aligned} \tag{6}$$

where τ symbolizes the temperature parameter and "$\cdot$" denotes the dot product operator. $\mathbb{I}_{\mathbf{Y}_{ij}=1}$ aims to find texts with the same label as that of the i-th text.

In these two applied components, ICL can provide beneficial information for the subsequent CCL task, while CCL can offer further guidance for the final classification task. Moreover, they can form a complementary relationship: ICL can provide a certain degree of variance in the obtained features, which can prevent the feature variability collapse. CCL has the capacity to mitigate class collision to a certain extent.

Classification Task: Finally, leveraging the abstract text features $\tilde{\mathbf{U}}$ derived from the CCL process, we perform the final classification task. We adopt an extra projection head $\Psi(\cdot)$ with the same structure as that of $\Phi(\cdot)$ to map the text embeddings $\tilde{\mathbf{U}}$ to another latent space, *i.e.*, $\mathbf{Q} = \Psi(\tilde{\mathbf{U}})$. Here, the dimension of $\mathbf{Q}$ is the number of classes. Then, we make predictions of these labeled data by performing a linear transformation followed by a ReLU activation on their hidden features. We specify the loss function as the commonly used cross-entropy function. Formally, the above operations can be expressed as:

$$\mathcal{L}^{\text{CE}} = -\frac{1}{|\mathcal{D}_{\text{train}}|} \sum_{i\in\mathcal{D}_{\text{train}}} \sum_{j}^{c} \mathcal{Y}_{ij} \log \mathbf{Q}_{ij} \tag{7}$$

where $\mathcal{D}_{\text{train}}$ is the set of training data from $\mathcal{D}_{\text{org}}$ and $\mathcal{Y}$ is the one-hot vector of the ground-truth label of the training data. c is the number of classes.

The adopted hierarchical architecture of tasks is similar to multi-task learning (Zhang and Yang 2021), which offers several advantages. First, by gradually progressing from ICL to CCL and then to classification, the model can extract more abstract and high-level features, which helps improve its generalization ability. Second, through step-by-step learning, the model can better utilize the data, as the results of the previous stage can provide better initialization and guidance for the subsequent stage. This inter-task correlation is beneficial for the model learning.

Model Optimation

Overall, the final loss function of our proposed model is the combination of the classification loss $\mathcal{L}^{\text{CE}}$, ICL loss $\mathcal{L}^{\text{ICL}}$, and CCL loss $\mathcal{L}^{\text{CCL}}$, which is formulated as follows:

$$\mathcal{L} = \mathcal{L}^{\text{CE}} + \eta\mathcal{L}^{\text{ICL}} + \zeta\mathcal{L}^{\text{CCL}}, \tag{8}$$

where η and ζ are hyperparameters that control the proportions of different losses.

During the model inference, we feed the obtained test text embeddings to the classification head $\Upsilon(\cdot)$ to evaluate the model performance.

Experiment

Datasets: We perform experiments on real-world STC datasets employed in earlier studies (Hu et al. 2019; Wang et al. 2021), *i.e.*, **Twitter**, **MR** (Pang and Lee 2005), **Snippets** (Phan, Nguyen, and Horiguchi 2008), **Ohsumed** (Hersh et al. 1994), and **TagMyNews** (Vitale, Ferragina, and Scaiella 2012). The statistics of these datasets are summarized in Table 1.

The preprocessing for these datasets is consistent with previous studies, including the removal of non-English characters, stop words, and infrequent words with counts of less than five. Following previous studies (Hu et al. 2019), we randomly sample 40 labeled short documents per class, half of which form the training set, and the other half form the validation set. The remaining data constitute the test set, and their labels are invisible during training.

Baselines: We select four types of baseline models for comparison. (1) Traditional models include **TF-IDF+SVM** and **PTE** (Tang, Qu, and Mei 2015). (2) Deep learning models contain **CNN**s (Kim 2014), **LSTM** (Liu et al. 2015) and **BERT** (Devlin et al. 2019). Here, **BERT-avg** and **BERT-cls** denote the text embeddings represented by the average word embeddings and the token CLS embedding, respectively.

Dataset	#Docs	#Train (ratio)	#Words	#Entities	#Tags	Avg.Len	#Classes
Twitter	10,000	40 (0.40%)	21,065	5,837	41	3.5	2
MR	10,662	40 (0.38%)	18,764	6,415	41	7.6	2
Snippets	12,340	160 (1.30%)	29,040	9,737	34	14.5	8
Ohsumed	7,400	460 (6.22%)	11,764	4,507	38	6.8	23
TagMyNews	32,549	140 (0.43%)	38,629	14,734	42	5.1	7

Table 1: Summary statistics of the evaluation datasets.

(3) GNN-based models consist of **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). (4) Deep short text models include **STCKA** (Chen et al. 2019), **HGAT** (Hu et al. 2019), **STGCN** (Ye et al. 2020), **SHINE** (Wang et al. 2021), **NC-HGAT** (Su et al. 2022), and **GIFT** (Liu et al. 2024b). Notably, we also provide several large language models, containing **GPT-3.5** (Ouyang et al. 2022), **Bloom-7.1B** (Scao et al. 2022), **Llama2-7B** (Touvron et al. 2023), and **Llama3-8B** (AI@Meta 2024). Due to computational resource constraints, we only fine-tune approximately 7B LLMs through some GPU reduction techniques.

Evaluation Metric: We use the accuracy (ACC) and macro-F1 score (F1) to evaluate the model performance, which are widely adopted by previous studies (Hu et al. 2019). All experiments are repeated ten times to obtain average metrics.

Result

Model Performance: Table 2 indicates that MI-DELIGHT achieves competitive performance across several datasets by a large margin in terms of accuracy and macro-F1 score. A key factor contributing to the remarkable superiority of MI-DELIGHT over other competing models lies in its deliberate design of a dual CL auxiliary task. This task serves the purpose of acquiring informative text representations and effectively capturing contrastive information at various levels. Specifically, the ICL and CCL within this framework enable the model to discern fine-grained details while also considering broader patterns and contexts. Moreover, the introduced hierarchical concept can help the model learn step by step, while fully utilizing the inter-task correlations, thereby enhancing the overall model performance. Moreover, we construct three types of graphs, including a word graph, a POS graph, and an entity graph, to incorporate statistical, linguistic and factual knowledge, which exploits semantic and syntactic information from the text and additional information from outside. All of the above operations are beneficial for better identifying the correct meanings of short texts.

We find that MI-DELIGHT basically achieves greater improvements on the Snippets, Ohsumed, and TagMyNews, which even considerably surpasses LLMs on these datasets. We attribute this phenomenon to the fact that the more unlabeled texts exist, the better MI-DELIGHT can extract useful self-supervised signals from them. LLMs have limited understanding in specific-domain (*e.g.*, medical domain) texts. However, LLMs achieve satisfactory performance in general texts such as Twitter and MR. This is because they are pretrained on vast amounts of high-quality data and have a large number of parameters. Moreover, they may have already encountered part of the test data.

Model Variants: To assess the effectiveness of each part of MI-DELIGHT, we design the following model variants to perform ablation experiments. (1) *w/o word graph*: We remove the word graph that introduces statistical knowledge. (2) *w/o POS graph*: We exclude the POS graph that incorporates linguistic knowledge. (3) *w/o entity graph*: We delete the entity graph that contains factual knowledge. (4) *w/o CCL and ICL*: We remove CCL and ICL simultaneously, leaving the text representation learning module which is combined with the cross-entropy loss for optimization. (5) *w/o CCL*: We eliminate the CCL module to demonstrate the role of the ICL module. (6) *w/o ICL*: We exclude the ICL module to confirm the efficiency of the CCL module. (7) *parallel*: We simply add projection heads for all tasks and perform them in parallel. We obtain several findings by observing the results presented in the first seven rows shown in Table 3. First, when we delete any part of the model, the performance of MI-DELIGHT decreases significantly, illustrating that each part plays an essential role in our model. Second, three constructed graphs used to enrich the short text information bring different types of information that play an indispensable role. Since the word graph can provide the most fundamental semantic information, removing it would significantly reduce the model's performance. Third, both the CCL and ICL modules are designed to allow the model to learn more discriminative text representations. Finally, our proposed hierarchical architecture is superior to the parallel version, since it fully utilizes the inter-task correlations.

Moreover, we explore the impacts of different approaches for generating positive sample pairs in the model. (1) **MI-DELIGHT (deletion)**: It randomly deletes a fraction of the words in a given sentence to generate an enhanced positive sample. (2) **MI-DELIGHT (context)**: It leverages pretrained large-scale language models (*e.g.*, BERT) to find a portion of the input text with suitable words for substitution. (3) **MI-DELIGHT (WordNet)**: It generates augmented positive pairs by replacing words of an input text with WordNet synonyms, which is the default experimental setting. The empirical results are shown in the last three rows of Table 3. As expected, the relevant metrics obtained across all the datasets drastically decrease when we adopt the deletion method for augmenting original texts. A plausible reason is deleting keywords from the original sentence changes its semantic information.

Model	Twitter		MR		Snippets		Ohsumed		TagMyNews	
	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1
TF-IDF+SVM	53.62	52.46	54.29	48.13	64.70	59.17	39.02	24.78	39.91	32.05
PTE	54.24	53.17	55.02	52.62	63.10	59.11	38.29	22.27	40.39	34.12
CNN	57.29	56.02	59.06	59.01	77.09	69.28	32.92	12.06	57.12	45.37
LSTM	60.28	60.22	60.89	60.70	75.89	67.72	28.86	7.20	57.32	45.56
BERT-avg	54.92	51.16	51.69	50.65	79.31	78.47	24.29	5.65	55.11	44.31
BERT-cls	52.06	43.41	53.50	47.02	81.55	79.06	22.26	5.50	58.19	42.35
TLGNN	59.02	54.56	59.22	59.36	70.25	63.29	35.76	13.12	45.25	33.52
HyperGAT	59.15	55.19	58.65	58.62	70.89	63.42	36.60	20.02	45.60	31.51
TextING	59.62	59.22	58.89	58.76	71.10	70.65	38.26	21.35	52.10	39.99
DADGNN	59.51	55.32	58.92	58.86	71.65	70.66	37.65	22.16	47.96	39.25
TextGCN	60.15	59.82	59.12	58.98	77.82	71.95	41.56	27.43	54.28	46.01
STCKA	57.56	57.02	53.25	51.19	68.96	61.27	32.20	12.25	32.15	23.26
HGAT	63.21	62.48	62.75	62.36	82.36	74.44	42.68	24.82	61.72	53.81
STGCN	64.33	64.29	58.25	58.22	70.01	69.93	35.22	28.30	35.65	35.16
SHINE	72.54	72.19	64.58	63.89	82.39	81.62	45.57	30.98	62.50	56.21
NC-HGAT	63.76	62.94	62.46	62.14	82.42	74.62	43.27	27.98	62.15	55.02
GIFT	73.16	73.16	65.21	65.21	83.73	82.35	45.62	31.25	63.26	56.92
Ours	**75.11**	**75.06**	**66.49**	**66.47**	**87.90**	**86.84**	**48.56**	**33.20**	**69.72**	**65.94**
GPT-3.5	81.23	80.02	87.43	86.62	66.52	63.48	47.98	32.49	61.43	54.79
Bloom-7.1B	87.52	86.56	87.03	86.96	71.39	60.76	37.46	30.12	66.13	62.13
Llama2-7B	87.45	86.43	87.26	86.69	73.05	68.11	42.16	30.19	67.31	64.33
Llama3-8B	89.50	89.47	84.38	84.16	61.68	60.62	38.76	22.93	65.80	60.69

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

Model	Twitter		MR		Snippets		Ohsumed		TagMyNews	
	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1
w/o word graph	62.60	62.10	55.02	54.96	76.15	74.80	29.08	21.26	62.52	57.19
w/o POS graph	72.40	71.65	65.46	65.25	85.86	85.19	46.96	28.19	66.39	59.10
w/o entity graph	70.42	70.36	64.76	64.79	85.95	85.12	47.55	29.98	67.28	62.90
w/o CCL and ICL	72.57	72.26	62.33	62.32	86.12	83.89	45.98	27.53	66.38	62.10
w/o CCL	72.19	72.18	66.02	65.92	85.34	83.99	48.38	31.59	66.56	62.45
w/o ICL	73.91	73.60	65.43	65.29	85.79	83.48	46.29	28.96	66.41	62.34
parallel	73.74	73.72	65.32	65.30	84.54	83.82	48.51	31.82	68.72	64.94
MI-DELIGHT (deletion)	72.04	71.92	63.52	63.50	83.98	82.26	44.59	27.92	67.44	62.67
MI-DELIGHT (context)	**75.66**	**75.56**	65.68	65.62	84.01	82.66	44.29	28.25	67.72	63.28
MI-DELIGHT (WordNet)	75.11	75.06	**66.49**	**66.47**	**87.90**	**86.84**	**48.56**	**32.20**	**69.72**	**65.94**

Table 3: The ablation and different text augmentation results (%) of various experimental settings.

Conclusion

In this work, we propose a novel model named MI-DELIGHT for STC. We build three types of graphs to introduce the statistical, linguistic, and factual information for enriching short texts. Also, we design a dual-level CL auxiliary tasks to capture multi-grained contrastive information. Moreover, we leverage a hierarchical structure to capture inter-task correlations. The empirical results reveal that MI-DELIGHT consistently outperforms other baselines, including some popular LLMs, across several datasets.

Code — https://github.com/KEAML-JLU/MI-DELIGHT

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

AI@Meta. 2024. Llama 3 Model Card.

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

Carlson, A.; Betteridge, J.; Kisiel, B.; Settles, B.; Jr., E. R. H.; and Mitchell, T. M. 2010. Toward an Architecture for Never-Ending Language Learning. In Fox, M.; and Poole, D., eds., *AAAI*.

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*.

Chang, Y.; Wang, X.; Wang, J.; Wu, Y.; Zhu, K.; Chen, H.; Yang, L.; Yi, X.; Wang, C.; Wang, Y.; et al. 2023. A survey on evaluation of large language models. *arXiv preprint arXiv:2307.03109*.

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

Chen, J.; Zhang, R.; Mao, Y.; and Xu, J. 2022. Contrastnet: A contrastive learning framework for few-shot text classification. In *AAAI*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*.

Di Stefano, L.; and Bulgarelli, A. 1999. A simple and efficient connected components labeling algorithm. In *ICIAP*.

Dilrukshi, I.; De Zoysa, K.; and Caldera, A. 2013. Twitter news classification using SVM. In *ICCSE*.

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*.

Edunov, S.; Ott, M.; Auli, M.; and Grangier, D. 2018. Understanding Back-Translation at Scale. In *EMNLP*.

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

Guan, R.; Liu, Y.; Feng, X.; and Li, X. 2021. VPALG: Paper-publication Prediction with Graph Neural Networks. In *CIKM*.

Hersh, W.; Buckley, C.; Leone, T.; and Hickam, D. 1994. OHSUMED: An interactive retrieval evaluation and new large test collection for research. In *SIGIR*.

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

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

Huynh, T.; Kornblith, S.; Walter, M. R.; Maire, M.; and Khademi, M. 2022. Boosting contrastive self-supervised learning with false negative cancellation. In *CVPR*.

Kenter, T.; and De Rijke, M. 2015. Short text similarity with word embeddings. In *CIKM*.

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*.

Kipf, T. N.; and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.

Li, M.; Liu, Y.; Giunchiglia, F.; Feng, X.; and Guan, R. 2024. Simple-Sampling and Hard-Mixup with Prototypes to Rebalance Contrastive Learning for Text Classification. *arxiv preprint arXiv:2405.11524*.

Li, Q.; Peng, H.; Li, J.; Xia, C.; Yang, R.; Sun, L.; Yu, P. S.; and He, L. 2022. A Survey on Text Classification: From Traditional to Deep Learning. *ACM TIST*, 13(2): 1–41.

Liu, N. F.; Gardner, M.; Belinkov, Y.; Peters, M. E.; and Smith, N. A. 2019. Linguistic Knowledge and Transferability of Contextual Representations. In *NAACL*.

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, P.; Qiu, X.; and Huang, X. 2016. Recurrent neural network for text classification with multi-task learning. In *IJCAI*.

Liu, P.; Qiu, X.; and Huang, X.-J. 2017. Adversarial Multi-task Learning for Text Classification. In *ACL*.

Liu, X.; You, X.; Zhang, X.; Wu, J.; and Lv, P. 2020. Tensor graph convolutional networks for text classification. In *AAAI*.

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.; Huang, L.; Cao, B.; Li, X.; Giunchiglia, F.; Feng, X.; and Guan, R. 2024a. A simple but effective approach for unsupervised few-shot graph classification. In *WWW*.

Liu, Y.; Huang, L.; Giunchiglia, F.; Feng, X.; and Guan, R. 2024b. Improved Graph Contrastive Learning for Short Text Classification. In *AAAI*.

Liu, Y.; Li, M.; Li, X.; Giunchiglia, F.; Feng, X.; and Guan, R. 2022. Few-shot node classification on attributed networks with graph meta-learning. In *SIGIR*.

Liu, Y.; Li, M.; Li, X.; Guan, R.; and Feng, X. 2023a. Local and Global: Temporal Question Answering via Information Fusion. In *IJCAI*.

Liu, Y.; Li, M.; Li, X.; Huang, L.; Giunchiglia, F.; Liang, Y.; Feng, X.; and Guan, R. 2024c. Meta-GPS++: Enhancing Graph Meta-Learning with Contrastive Learning and Self-Training. *ACM TKDD*, 18(9): 1–30.

Liu, Y.; Li, M.; Liang, D.; Li, X.; Giunchiglia, F.; Huang, L.; Feng, X.; and Guan, R. 2024d. Resolving Word Vagueness with Scenario-guided Adapter for Natural Language Inference. In *IJCAI*.

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*.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training language models to follow instructions with human feedback. In *NeurIPS*.

Pan, L.; Hang, C.-W.; Sil, A.; and Potdar, S. 2022. Improved text classification via contrastive adversarial training. In *AAAI*.

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 *WWW*.

Scao, T. L.; Fan, A.; Akiki, C.; Pavlick, E.; Ilić, S.; Hesslow, D.; Castagné, R.; Luccioni, A. S.; Yvon, F.; et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

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*.

Thilakaratne, M.; Falkner, K.; and Atapattu, T. 2019. A systematic review on literature-based discovery: general overview, methodology, & statistical analysis. *CSUR*, 52(6): 1–34.

Tian, Y.; Krishnan, D.; and Isola, P. 2020. Contrastive multiview coding. In *ECCV*.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Vitale, D.; Ferragina, P.; and Scaiella, U. 2012. Classification of short texts by deploying topical annotations. In *ECIR*.

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*.

Wei, J.; and Zou, K. 2019. EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. In *EMNLP-IJNLP*.

Wu, X.; Gao, C.; Zang, L.; Han, J.; Wang, Z.; and Hu, S. 2022. Esimcse: Enhanced sample building method for contrastive learning of unsupervised sentence embedding. In *COLING*.

Xie, Z.; Wang, S. I.; Li, J.; Lévy, D.; Nie, A.; Jurafsky, D.; and Ng, A. Y. 2017. Data noising as smoothing in neural network language models. In *ICLR*.

Yang, T.; Hu, L.; Shi, C.; Ji, H.; Li, X.; and Nie, L. 2021. HGAT: Heterogeneous graph attention networks for semi-supervised short text classification. *ACM TOIS*, 39(3): 1–29.

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*.

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, Y.; and Yang, Q. 2021. A survey on multi-task learning. *IEEE TKDE*, 34(12): 5586–5609.

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*.

Zheng, M.; Wang, F.; You, S.; Qian, C.; Zhang, C.; Wang, X.; and Xu, C. 2021. Weakly supervised contrastive learning. In *CVPR*.