Expert Systems With Applications 266 (2025) 125952

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Contrastive multi-graph learning with neighbor hierarchical sifting for semi-supervised text classification

Wei Ai [a], Jianbin Li [a], Ze Wang [a], Yingying Wei [a], Tao Meng [a],*, Keqin Li [b]

[a] College of Computer and Mathematics, Central South University of Forestry and Technology, Changsha, Hunan 410004, China
[b] Department of Computer Science, State University of New York, New Paltz, NY 12561, USA

ARTICLE INFO

Keywords:
Text classification
Graph neural network
Graph contrastive learning
Negative sample selection

ABSTRACT

Graph contrastive learning has been successfully applied in text classification due to its remarkable ability for self-supervised node representation learning. However, explicit graph augmentations may lead to a loss of semantics in the contrastive views. Secondly, existing methods tend to overlook edge features and the varying significance of node features during multi-graph learning. Moreover, the contrastive loss suffer from false negatives. To address these limitations, we propose a novel method of contrastive multi-graph learning with neighbor hierarchical sifting for semi-supervised text classification, namely ConNHS. Specifically, we exploit core features to form a multi-relational text graph, enhancing semantic connections among texts. By separating text graphs, we provide diverse views for contrastive learning. Our approach ensures optimal preservation of the graph information, minimizing data loss and distortion. Then, we separately execute relation-aware propagation and cross-graph attention propagation, which effectively leverages the varying correlations between nodes and edge features while harmonizing the information fusion across graphs. Subsequently, we present the neighbor hierarchical sifting loss (NHS) to refine the negative selection. For one thing, following the homophily assumption, NHS masks first-order neighbors of the anchor and positives from being negatives. For another, NHS excludes the high-order neighbors analogous to the anchor based on their similarities. Consequently, it effectively reduces the occurrence of false negatives, preventing the expansion of the distance between similar samples in the embedding space. Our experiments on ThuCNews, SogouNews, 20 Newsgroups, and Ohsumed datasets achieved 95.86%, 97.52%, 87.43%, and 70.65%, which demonstrates competitive results in semi-supervised text classification.

1. Introduction

Text classification is a crucial task in natural language processing, with a wide range of applications, including sentiment analysis, news categorization, question-answering systems, and spam filtering. Traditional deep learning methods (Chang et al., 2020; Lai et al., 2015; Shi et al., 2024; Tai et al., 2015) approach text as a complete whole and capture features from locally continuous word sequences, achieving significant strides. Recent advances in graph-based methods (Linmei et al., 2019; Piao et al., 2022; Yang, Miao, et al., 2022; Yao et al., 2019) have ushered in a new era of text classification, leveraging the ability of Graph Neural Networks in generating node representations to drive competitive performance.

The first step for graph-based text classification tasks is to break the independence of different data samples by constructing graph topologies for unconnected free texts. The second step involves leveraging the ability of graph neural networks to capture both global and local information to learn text representations. Specifically, existing methods (Lei et al., 2021; Lin et al., 2021; Yao et al., 2019; Zhang & Zhang, 2020) treat words and documents as nodes and construct a heterogeneous text graph based on the point-wise mutual information (PMI) relationships between words and the TF-IDF relationships between words and documents. Despite such methods having achieved promising results, they neglect the rich and deep semantics, which is pivotal for capturing the core intent of the text. To account for deep textual semantics, some studies (Li et al., 2021; Liu et al., 2020) propose to construct multi-typed text graphs (i.e., semantic, syntactic, and sequential contexts). TensorGCN (Liu et al., 2020) executes GCN propagation within different text graphs separately to aggregate neighboring information of nodes. Subsequently, to integrate across-graph features, a virtual graph for nodes at the same positions is constructed to perform inter-graph propagation. TextGTL (Li et al., 2021) designs a

* Corresponding author.
E-mail addresses: aiwei@hnu.edu.cn (W. Ai), jianbinli@csuft.edu.cn (J. Li), zewang@csuft.edu.cn (Z. Wang), yingying.wei@csuft.edu.cn (Y. Wei), mengtao@hnu.edu.cn (T. Meng), lik@newpaltz.edu (K. Li).

https://doi.org/10.1016/j.eswa.2024.125952
Received 17 July 2024; Received in revised form 16 October 2024; Accepted 25 November 2024
Available online 9 December 2024

two-layer parallel GCN to learn document node representations across graphs. Specifically, it independently aggregates information over multiple graphs in the first layer. Then, it performs average pooling on the outputs of the different graphs from the first layer to serve as the input for the second layer. However, these methods have some drawbacks. Firstly, they perform average pooling aggregation on neighboring nodes during intra-graph propagation, neglecting the edge features and the varying relevance between nodes. Secondly, they assign equal weights to different features during the inter-graph propagation, ignoring the intrinsic differences inherent in these features. Overall, the current works neither construct relationships between texts using rich semantics nor propose an effective method for node representation learning across multiple graphs. These shortcomings indicate that exploring a text classification method capable of enhancing semantic connections between texts and improving the multi-graph learning process remains an unresolved challenge.

The emergence of a large amount of unlabeled text has made semi-supervised text classification extremely challenging. Recently, some studies (Li et al., 2023; Sun et al., 2022; Zhao & Song, 2023) have leveraged the self-supervised representation learning capabilities of graph contrastive learning (GCL) to mitigate the issue of label scarcity in text classification. However, these methods rely on explicit graph augmentation to obtain contrastive views. This not only requires prior domain knowledge or trial and error to determine optimal graph augmentation parameters but also may fail to preserve the integrity of task-relevant information through augmentation. Specifically, common augmentation techniques like randomly deleting document nodes or key edges (Lan et al., 2023) can significantly alter the meaning of the text. This reduces the consistency of learnable representations between contrastive views, thereby misleading the learning process of graph neural networks. Moreover, the fundamental goal of GCL is to design an appropriate contrastive loss function to cluster similar nodes while separating dissimilar nodes. However, current methods like CGA2TC (Yang, Miao, et al., 2022) typically employ the NT-Xent contrastive loss function, which is widely used in GCL. Such contrastive loss function considers nodes at the same position as positive samples while treating the remaining nodes within and across views as negative samples. This inevitably leads to selecting document nodes with similar semantics as negative samples and results in similar nodes being far apart in the latent space, which contradicts the fundamental goal of GCL. Existing GCL-based text classification methods result in incomplete information due to their dependence on graph augmentation and produce false negatives on the ground that the use of common contrastive loss. These shortcomings underscore the necessity of developing a novel augmentation-free contrastive learning framework, aimed at overcoming information loss and false negative issues.

To tackle the aforementioned challenges, we propose a novel method of **Con**trastive multi-graph learning with **N**eighbor **H**ierarchical **S**ifting for semi-supervised text classification, named ConNHS. The proposed method eliminates the need for explicit graph augmentation and introduces a novel contrastive loss function to optimize representation learning. Firstly, we extract titles, keywords, and events to construct a multi-relational text graph that can represent more latent semantic connections. Secondly, to avoid the loss of structural information caused by graph augmentation, we separate the multi-relational text graph to derive semantic subgraphs (corresponding to titles, keywords, and events). This provides multiple views for the graph contrastive learning stage. Subsequently, we propose a relation-aware graph convolutional network (RW-GCN) to perform intra-graph propagation within each semantic subgraph, which considers the varying correlations between document nodes and incorporates edge feature information. Moreover, considering the differences among semantic subgraphs, we design a cross-graph attention network (CGAN) for inter-graph propagation to obtain fused node representations, effectively harmonizing the feature information from different subgraphs. Additionally, we present the neighbor hierarchical sifting loss (NHS) to circumvent the false negative pairs that could undermine contrastive learning efforts. Specifically, NHS masks the first-order neighbors of the anchor, as the construction of the multi-relational text graph is dependent on the homophily assumption, i.e., connected document nodes tend to share the same label. Furthermore, NHS draws signals from the similarity score matrix of the fused node representations, excluding high-order neighbors with high similarity to the anchor from being chosen as negatives. This dual approach, rooted in graph structure and node attributes, prevents similar nodes from being distanced in the latent space. Finally, we input the fused node representations obtained from multiple subgraphs into a logistic regression classifier to achieve the final classification results.

The main contributions of this article can be summarized as follows:

- We harness core features to forge a multi-relational text graph that contains multiple semantic connections among documents. Meanwhile, we propose RW-GCN to leverage edge features and capture varying correlations between nodes. We also design CGAN to coordinate the fusion of feature information across graphs.
- We propose a contrastive learning method for semi-supervised text classification that does not require graph augmentation. Our innovative contrastive loss function effectively optimizes negative selection and avoids the occurrence of false negatives, thus providing clearer clustering boundaries for downstream text classification.
- We test the proposed method on four real-world datasets (including Thucnews, Sogounews, 20NG, and Ohsumed), and the results demonstrate the effectiveness of ConNHS for semi-supervised text classification tasks.

The rest of this paper is organized as follows: Section 2 introduces related work, Section 3 presents the detailed method, Section 4 gives the experimental setup and results, and finally, Section 5 gives a brief conclusion.

2. Related work

2.1. Deep learning for text classification

In the early stages of text classification, methods primarily focused on machine learning-based techniques, heavily relying on feature engineering dependent on specific domain knowledge and experience. With the advent of deep learning models, the need for feature engineering has significantly been alleviated, as these models possess the capability to learn textual features automatically. Specifically, TextCNN (Kim, 2014) is the first attempt to transfer the CNN model, widely applied in computer vision, to text classification tasks. It extracts local features using multiple filters, but this method struggles to capture long-range dependencies in text sequences effectively. RNN-based methods, such as TopicRNN (Dieng et al., 2016) and RNN-Capsule (Wang et al., 2018), can address the long-term dependency problem and learn more comprehensive text representations, but they may encounter issues like gradient explosion or vanishing gradients. In recent years, Transformer-based pre-trained models (Kenton & Toutanova, 2019; Liu et al., 2019; Shi et al., 2024), with their exceptional semantic understanding capabilities, have been widely applied in text classification tasks, achieving significant results. Despite the remarkable success of sequence-based deep learning models in text classification tasks, they still exhibit certain inherent limitations. For example, they primarily focus on token-level information processing, potentially overlooking the complex intertextual relationships and higher-level semantic structures. These limitations highlight the necessity of exploring text classification methods with enhanced semantic understanding capabilities to better address complex classification challenges.

2.2. GNN for text classification

Graph Neural Networks (GNN) are deep learning models designed for graph-structured data. They are widely used in fields such as social network analysis, recommendation systems, and molecular chemistry. GNNs leverage the information from nodes and edges in a graph to effectively represent and learn from graph data. The Graph Convolutional Network (GCN) (Kipf & Welling, 2016) model achieves good results by performing spectral convolutions on node features, making it widely applicable for tasks like node classification and graph embedding. The Graph Attention Network (GAT) (Veličković et al., 2018) introduces an attention mechanism that allows the model to assign different weights when aggregating information from neighboring nodes, enhancing the model's expressive power and flexibility. With the rapid development of graph neural networks, a variety of graph-based text classification models have also emerged.

These models can be broadly categorized into document-level and corpus-level types. Document-level methods treat words as nodes and construct an independent text graph for each document, effectively mining contextually relevant word relationships. For example, Text-Level-GNN (Huang et al., 2019) uses a sliding window approach, employing a limited number of nodes and edges in each text graph to reduce memory and computational overhead. Meanwhile, TextFCG (Wang et al., 2023) builds a single graph for all words in a text, marking edges with various contextual relationships, and adopts GNN and GRU for text classification. On the other hand, corpus-level methods capture the global structural information of a corpus by constructing one or more graphs containing both word and document nodes, which include various relationships like word-word and word-document. TextGCN (Yao et al., 2019) constructs the entire corpus as a heterogeneous graph, using word nodes as intermediaries for information transfer and facilitating inter-document information exchange through a two-layer GCN. TensorGCN (Liu et al., 2020) constructs a text graph tensor to capture semantic, syntactic, and sequential contextual information and uses both intra-graph and inter-graph propagation to harmonize heterogeneous information from multiple graphs. However, these methods often fall short in fully capturing textual semantic information when constructing graph structures, leading to an inadequate understanding of the deeper meanings within the text. Additionally, when employing multi-type text graphs, these approaches face challenges in learning both intra and inter graphs due to feature discrepancies between different types of nodes. This inconsistency in features can hinder the model to accurately grasp global semantic relationships and effectively propagate information.

2.3. Graph contrastive learning

Graph contrastive learning (Mo et al., 2022; Xia et al., 2022; Xu et al., 2021; Yang, Chen, et al., 2022) is a technique for extracting features efficiently using unlabeled data. Its core idea is to generate positive and negative samples by transforming the original data, thereby reducing the distance between similar data and increasing the distance between dissimilar data in the feature space, achieving a clustering-like effect. The process of graph contrastive learning mainly covers three key stages. The first is the graph data augmentation stage, which is crucial to ensure the difference and diversity between views and has a significant impact on the final model's performance. Second is embedding learning, which involves encoding node samples to generate contrastive samples. Finally, the calculation of contrastive loss includes defining positive and negative sample pairs, thereby promoting the model to learn more discriminative node features.

Recently, in the field of graph contrastive learning, numerous efficient methods and applications have gradually emerged. For instance, MVGRL (Hassani & Khasahmadi, 2020) employs graph diffusion techniques for graph-level data augmentation on the original input graph, thereby obtaining views containing richer global information. GraphCL (You et al., 2020), on the other hand, explores various graph augmentation strategies to address the heterogeneity issue in graph data. Simultaneously, GCA (Zhu et al., 2021) introduces an adaptive data augmentation scheme, moving away from the traditional practice of uniformly dropping edges or perturbing features. Instead, it emphasizes the enhancement of essential nodes and edges and the disruption of node features to obtain more effective views. GCNSS (Miao et al., 2022) effectively mitigates the false negative pairs problem in graph contrastive learning by utilizing label information. Additionally, NCLA (Shen et al., 2023) proposes a new learnable graph augmentation strategy, generating higher-quality contrastive views. For GCL-based text classification methods, ConKGNN (Lan et al., 2023) constructs a unified graph that includes text and related knowledge graph (KG) information and introduces contrastive learning to accomplish the text classification task. However, the random graph augmentation it utilizes can lead to unpredictable information loss. TextGCL (Zhao & Song, 2023) simultaneously trains GCN and BERT, utilizing contrastive learning loss to learn precise text representations. It lacks a discerning mechanism in the selection of negative samples, inevitably introducing false negatives.

3. Proposed method

In this section, we first provide a brief overview of our proposed ConNHS method, followed by a detailed explanation of its constituent modules. The overall process of ConNHS is illustrated in Fig. 1. As shown, our proposed ConNHS comprises five main stages: (1) Feature extraction: For semantically enriched texts, we start from the semantic level by extracting the titles, keywords, and events of the texts. These core features are used as the basis for constructing the text graph. (2) Multi-relational text graph construction: Inspired by the intrinsic logic that humans use to classify texts, we construct multiple document-to-document relationships by calculating the similarity of core features in the embedding space. The constructed text graph contains more latent semantic connections between document nodes. (3) Multi-graph learning: To avoid explicit graph augmentation, we separate the multi-relational text graph into different semantic subgraphs. We propose a relation-aware graph convolutional network to perform intra-graph propagation within each subgraph. This method fully considers edge features and the varying correlations between nodes, thus aggregating more significant neighborhood information. Additionally, given the differences in node features across subgraphs, we design a cross-graph attention network. It facilitates inter-graph propagation to obtain the fused text representations, thereby enabling a comprehensive and nuanced understanding of textual congruence. (4) Contrastive learning with NHS: To acquire precise text representations, we apply a novel graph contrastive learning methodology for model training. By presenting an innovative contrastive loss to refine negative selection, the ubiquitous quandary of false negatives in GCL is substantially mitigated, thereby enhancing the fidelity and robustness of the text representations. (5) Label prediction: We leverage the pre-trained model to obtain fused text representation and input them into a logistic regression classifier to predict the label of each text. In the subsequent sections of the paper, we will describe each component of the ConNHS model in detail.

3.1. Feature extraction

The fundamental logical judgment for humans to ascertain the domain of a text is recognizing the features that can represent the core intention of the text. For instance, when the word "goalkeeper" is present in two pieces of news, our cognitive systems are inclined to categorize both texts under the sports domain. The inclination is rooted in the understanding that texts sharing similar snippets of information or vocabulary are likely to emanate from the same sphere. Drawing inspiration from this human-centric logic for classifying texts, we aim

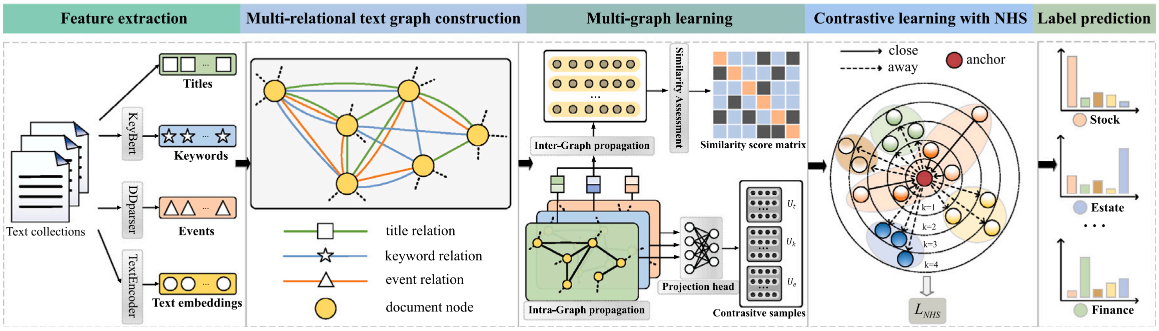

Fig. 1. Flow chart of the proposed ConNHS. Initially, we construct a multi-relational text graph by leveraging inherent core features (titles, keywords, events) to establish semantic connections among texts while encoding textual content as initial node representations. Subsequently, relational separation yields distinct subgraphs, upon which intra-graph and inter-graph propagation are performed to obtain contrastive samples and similarity score matrix. During Contrastive learning with NHS, negative selection is optimized to encourage more explicit cluster boundaries (minimizing intra-class distances while maximizing inter-class distances; distinct colors indicate different clusters). Ultimately, predicted labels are assigned to document nodes via a logical classifier.

to extract various core features to forge links between texts that are otherwise unconnected.

Title: Serving as the introductory sentence of an article, titles typically encapsulate information pertinent to the topic of text, providing a high-level synopsis of the content. Fundamentally, the title is constructed as the first sentence imbued with comprehensive semantic information, necessitating no further processing. The titles set can be formalized as $Title = \{t_1, t_2, \dots, t_n\}$, where t_i is the title of the text i.

Event: Moreover, we incorporate the concept of events to achieve a more sophisticated level of textual representation. Typically, an event is characterized as an action or condition that has transpired or is currently happening. Utilizing events as a means of text representation is a widely acknowledged approach, offering a clearer conveyance of textual information than mere sentences or phrases. Therefore, considering that events are mainly composed of objects and the actions they emit, we introduce the definition of event (Zhang et al., 2022) as follows:

$$Event = (W, C, O), \tag{1}$$

where W represents the action that occurs during the event, C is the factor that causes the event to happen, and O is another object that is involved in the event. The main task of event extraction is identifying and extracting the subject, action, and object. We choose DDparser (Zhang et al., 2020) and Stanza (Zhang et al., 2021) as extraction tools to extract events from Chinese and English texts, respectively. The events can be formalized as $EventSet = \{E_1, E_2, \dots, E_n\}$, where E_i is the set of events extracted from the text i.

Keyword: Events distil the essence of a text under the assumption that its semantic core is anchored in specific paragraphs or sentences. Nonetheless, in instances where the content is more scattered, particularly in lengthier texts, the event-centric approach might fall short of capturing textual semantics at the granularity of individual words. Consequently, to address this granularity gap and ensure a comprehensive understanding of the textual thematic breadth, we establish semantic relationships between texts based on keywords at a more fine-grained level. We choose KeyBert[1] as the extraction tool, and the extracted keywords are formalized as $KeywordSet = \{K_1, K_2, \dots, K_n\}$, where K_i is the set of keywords extracted from the text i.

The core features delineated above and text contents are transmuted into a computable format via a text embedding model, with the preeminent choice being models pre-trained on extensive corpora. The preference is rooted in two fundamental advantages: firstly, pre-trained models are imbued with a robust knowledge base, endowing them with superior semantic comprehension capabilities; secondly, these models exhibit context sensitivity, which is crucial for adeptly navigating the complexities of homographs—words identical in spelling but divergent in meaning. With the aim of precisely representing the core features and textual contents, this study will employ a text encoder that is comprised of the LangChain[2] framework and BGE-M3 (Chen et al., 2024), a variant version of Bert. This combination is tasked with converting each title, keyword, and event into vector representations, which are instrumental in constructing a multi-relational text graph and laying the groundwork for intricate semantic relationships between texts.

3.2. Multi-relational text graph construction

A common graph construction strategy for graph-based text classification methods (Yao et al., 2019) involves analyzing the PMI relationships between words and the TF-IDF relationships between words and documents. This approach, however, overlooks the deep semantic information that can represent the underlying relationships within the text. Therefore, we calculate the semantic similarity between the extracted features to facilitate the construction of multiple semantic relationships between document nodes, corresponding to title relationships, keyword relationships, and event relationships. Based on the rich features inherent in the text, the constructed text graph can maximize the connections between similar documents. Formally, considering the multi-relational text graph as: $G = \{V, A, R\}$ contains document nodes and relationship collection, where $V = \{v_1, v_2, \dots, v_n\}$, v_i represents the document node i, and n represents the number of document nodes. Moreover, A is the adjacency matrix of the text graph. The edge sets are represented by $R = \{T, K, E\}$, corresponding to the title, keyword, and event relationship.

Node representation: The majority of texts are interspersed with information unrelated to the main topic, underscoring the necessity for meticulous preprocessing of text content within the source space. For example, the primary body of news articles often encompasses author signatures, names of news agencies, and additional elements that are tangential to the core intention of the article. The process of obtaining the initial node representation is as follows:

$$m_i = TextEncoder(c_i), \tag{2}$$

where c_i is the preprocessed content of text i, $m_i \in \mathbb{R}^d$, and d is the dimension of node representation.

Title relation: Titles serve as succinct summaries of textual content and are pivotal in the classification of texts. It is observed that texts belonging to the same category often exhibit a notable similarity in their titles. To capitalize on this observation, we introduce a scoring mechanism designed to quantify the similarity between titles. The

[1] https://github.com/MaartenGr/KeyBERT

[2] https://github.com/langchain-ai/langchain

quantification of the semantic similarity between titles t_i and t_j can be expressed in the following manner:

$$S_{ij}^t = Sim(t_i, t_j), \tag{3}$$

where $Sim(\cdot, \cdot)$ denotes the cosine similarity measure, which quantifies the magnitude of the angle formed by two vector representations X and Y in the latent space. It can be formulated as:

$$Sim(X, Y) = \frac{\sum_{i=1}^{n}(x_i \cdot y_i)}{(\sum_{i=1}^{n} x_i^2)^{\frac{1}{2}} \cdot (\sum_{i=1}^{n} y_i^2)^{\frac{1}{2}}}, \tag{4}$$

for the title relation between text i and text j, we define it as follows:

$$R_{ij}^t = \begin{cases} 1 & \text{if } S_{ij}^t > \rho_t, \\ 0 & \text{otherwise}, \end{cases} \tag{5}$$

if the quantified semantic similarity S_{ij}^t between titles t_i and t_j transcend the predefined threshold ρ_t, the title relation R_{ij}^t shall be established.

Event relation: Events describe the core intent of a document and thus can serve as a significant feature in constructing the potential connections of texts. Different documents often contain multiple events. Two events sharing a similarity score exceeding the pre-determined threshold ρ_e are considered as a matching event pair. For text i and text j, we quantify the relatedness of their constituent events as follows:

$$L_{ij}^e = \{(e_a, e_b) | e_a \in E_i, e_b \in E_j, Sim(e_a, e_b) > \rho_e\}, \tag{6}$$

where L_{ij}^e is the list of matching event pair.

$$R_{ij}^e = \begin{cases} 1 & \text{if } Counter(L_{ij}^e) > \gamma_e, \\ 0 & \text{otherwise}, \end{cases} \tag{7}$$

where $Counter(\cdot)$ is a utility function that serves to count the elements within a list. If the matching event pairs shared by text i and text j exceed the minimum association coefficient γ_e, the event relation R_{ij}^e will be established.

Keyword relation: Keywords are vital in understanding the theme of a text, offering a new perspective for establishing semantic relations between text nodes. The keywords exhibiting a similarity score that surpasses the predetermined threshold ρ_k are deemed to be a matching keyword pair. The procedure for establishing keyword relation bears a resemblance to that for event relation. It can be formulated as follows:

$$L_{ij}^k = \{(k_a, k_b) | k_a \in K_i, k_b \in K_j, Sim(k_a, k_b) > \rho_k\}, \tag{8}$$

where L_{ij}^k is the list of matching keyword pair.

$$R_{ij}^k = \begin{cases} 1 & \text{if } Counter(L_{ij}^k) > \gamma_k, \\ 0 & \text{otherwise}, \end{cases} \tag{9}$$

if the number of matching keyword pairs is greater than the minimum association coefficient γ_k, the keyword relation R_{ij}^k shall be instantiated.

Titles, keywords, and events serve as foundational elements in constructing connections between texts, each offering a unique perspective on the features that define their semantic relationships. The multifaceted approach enables texts that are potentially analogous to share information across their respective nodes, thereby facilitating a more enriched and nuanced text representation learning.

3.3. Multi-graph learning

Recent studies have proposed constructing multi-typed text graphs for text classification tasks, but they have limitations during multi-graph learning. Firstly, they discount the edge features and use average pooling to aggregate neighborhood information during the intra-graph propagation. This aggregation method assumes that all neighboring document nodes are equally important, disregarding the diversity of documents. Secondly, they overlook the differences in node features across different text graphs during inter-graph propagation.

To maintain the integrity of task-relevant graph structural information while providing diverse views for graph contrastive learning, a crucial step is separating the multi-relational text graph according to the relationship type, leading to the creation of semantic subgraphs, as illustrated in Fig. 1. After that, we perform intra-graph and inter-graph propagation on these derived semantic subgraphs.

Intra-graph propagation: Rather than conventional GCN (Kipf & Welling, 2016), we propose a relation-aware graph convolution network which consists of a relation-aware aggregation operator $g(\cdot;\theta_g)$ and a transformation operator $f(\cdot;\theta_f)$. In detail, let x_i denote the feature representation of node v_i at the lth layer, the aggregation operation can be formally expressed as follows:

$$g(\cdot;\theta_g) = \sum_{x_j^l \in \mathcal{N}(x_i^l)} h(x_j^l - x_j^l;\theta_h) \cdot (x_j^l - x_j^l), \tag{10}$$

where $h(\cdot;\theta_h)$ represents a learnable function parameterized by θ_h, whose purpose is to ascertain the important weights quantifying the correlation between document nodes. The instantiation of $h(\cdot;\theta_h)$ is achieved through a fully connected layer followed by a sigmoid activation. Let $\mathcal{N}(x_i^l)$ denote the feature set of neighboring nodes x_i^l at the lth layer, wherein x_j^l corresponds to the feature representation of the neighbor node v_j. Notably, the edges $x_j^l \in \mathcal{N}(x_i^l)$, $(x_j^l - x_i^l)$ connecting the centroid node and its neighboring nodes serve as input to the aggregation operator. In other words, $h(x_j^l - x_i^l;\theta_h)$ can be interpreted as the importance weights characterizing the relation between x_j^l and x_i^l. Furthermore, we aggregate all the weighted correlation edge features as the aggregating features in the graph, consequently capturing the latent relations among diverse document nodes. Concerning the transformation operator $f(\cdot;\theta_f)$, we concatenate the node feature x_i^l with the aggregating features obtained from $g(\cdot;\theta_g)$ as its input. The updated feature x_i^{l+1} of node v_i by the RW-GCN at the $(l+1)^{th}$ layer can be formally defined as follows:

$$x_i^{l+1} = f([x_i^l, g(\cdot;\theta_g)];\theta_f), \tag{11}$$

where $x_i^{l+1} \in \mathbb{R}^{2 \times d}$, $[\cdot, \cdot]$ is a concatenation operation. θ_f is the independent learnable weight matrix to transform the input features.

Within the realm of graph contrastive learning, a conventional strategy involves augmenting the input graph to generate two distinct views, followed by the extraction of feature representations from these views using a graph encoder. This methodology, reliant on graph augmentation, presents two primary challenges: Firstly, prevalent graph augmentation techniques, such as edge dropping and attribute masking, risk compromising the structural integrity and semantic content of the graph. For example, the elimination of critical edges could adversely affect the learning of node representations. Secondly, the application of graph augmentation techniques typically necessitates iterative fine-tuning to identify optimal parameters, a process that can be both time-consuming and imprecise. In response to these issues, our method obviates the necessity for intricate graph augmentation procedures. Instead, we employ relation-aware GCN to process semantic subgraphs which inherently possess distinct adjacencies. This approach enables the derivation of varied and diverse views without the introduction of graph augmentations, thereby preserving the original graph structural and semantic integrity.

Intra-graph propagation: After intra-graph propagation, each document node learns unique feature information under different semantic relationships. Therefore, we design a cross-graph attention network to coordinate and integrate diverse feature information. The process of aggregating document node representations from different subgraphs can be formalized as follows:

$$\alpha_r = softmax(k^T tanh(p(x_{i,r};\theta_p))), \tag{12}$$

where $x_{i,r}$ is the representation of the document node v_i at the relation r subgraph, and α_r is the attention weight. $p(\cdot;\theta_p)$ is a feedforward neural network parameterized by θ_p. Then, we use the computed attention weights to perform cross-graph information propagation. The process is as follows:

$$H_i' = (\otimes\{\alpha_r \cdot x_{i,r}\}|_{r=1}^{R}), \tag{13}$$

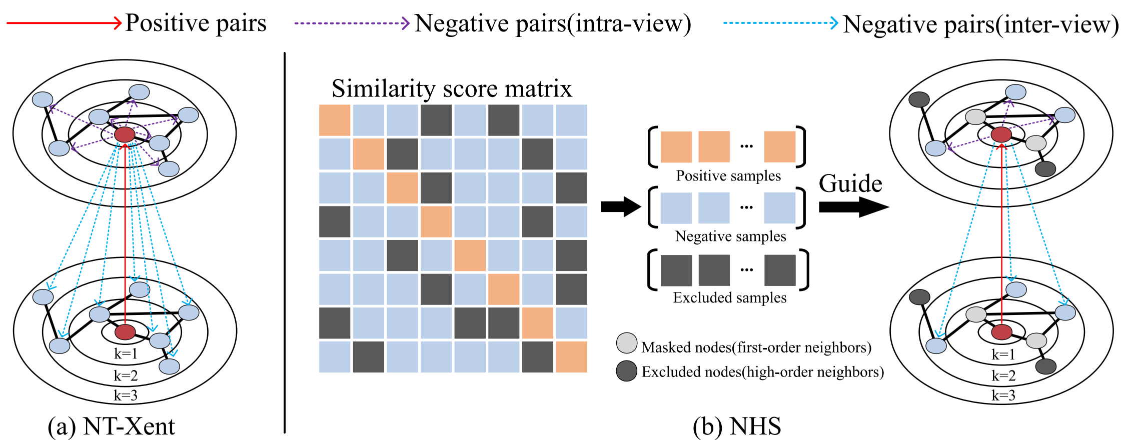
Positive pairs
Negative pairs(intra-view)
Negative pairs(inter-view)
Similarity score matrix
Positive samples
Negative samples
Excluded samples
Guide
Masked nodes(first-order neighbors)
Excluded nodes(high-order neighbors)
k=1
k=2
k=3
(a) NT-Xent
(b) NHS

Fig. 2. Definition of negative pairs in different contrastive losses. Fig. 2 showcases different negative selection definition strategies. Specifically, both NT-Xent and NHS recognize nodes positioned identically across views as positive samples for the anchor. However, NT-Xent designates all remaining nodes as negatives. In contrast, NHS masks first-order neighbors of the anchor document node and the positive nodes based on the graph homophily principle, and also, based on the similarity score matrix of fused node representations, as shown in (b), it excludes those high-order neighbors that exhibit significant similarity to the anchor. To facilitate a more straightforward interpretation, sifted hierarchical neighbors that will not be included in the contrastive learning process are indicated with specific colors in (b).

where $\otimes$ is the sum operator, and $R = \{title, keyword, event\}$ is the set of semantic relations. H_i' is the fused representation of the document node v_i.

Projection mapping: To mitigate the impact of irrelevant features across contrasting views while preserving the most salient information, certain graph contrastive learning approaches advocate mapping node embeddings onto a specific latent space. Consistent with prior approaches, we employ a projection head $q(\cdot;\theta_q)$ to transform the node embedding representation into a tailored latent space prior to computing the contrastive loss objective. In this paper, the mapping process can be formulated as follows:

$$u_i = q(h_i;\theta_q), \tag{14}$$

where h_i is the learned representation of document node v_i, and u_i is the mapping result, which will be regarded as contrastive node samples for contrastive learning.

3.4. Contrastive learning with NHS

A key step in graph contrastive learning is designing an appropriate contrastive loss function to cluster similar nodes while separating dissimilar nodes. In traditional contrastive learning paradigms, the contrastive loss NT-Xent typically selects nodes at corresponding positions across views as positive samples for the anchor node. Conversely, all remaining nodes, irrespective of their positioning within or across views, are designated as negative samples. However, such negative selection that NT-Xent adopts inevitably induces false negative pairs. It inadvertently broadens the gap between nodes that are inherently similar, thereby contravening the foundational goal of GCL.

The fundamental principle underlying contrastive learning can be generally encapsulated as employing a transformation function $f(\cdot)$ to map the input node representation x onto $f(x)$, such that the resultant mapping adheres to the following inequality constraint:

$$Dist(f(x_i), f(x_i^+)) \ll Dist(f(x_i), f(x_i^-)), \tag{15}$$

where x_i^+ denotes a positive sample exhibiting similarity to x_i, while x_i^- represents a negative sample dissimilar to x_i. The function $Dist(\cdot,\cdot)$ serves as a similarity measure employed to quantify the degree of similarity between the node embedding representations.

Neighbor Hierarchical Sifting: To address this challenge, our work proposes the neighbor hierarchical sifting loss designed to prevent the incidence of false negative pairs generation, as illustrated in Fig. 2. In keeping with the conventional loss for identifying positive pairs, we continue to regard nodes situated in matching positions across views as positive samples relative to each other. Importantly, extending our consideration to the graph homophily, not only the first-order neighbors of the anchor node but also positive nodes across different views are masked and removed from the negatives. Furthermore, for high-order neighboring nodes that belong to the same category yet lack direct connections, their selection as negative samples can also compromise contrastive learning efficacy. To address this, we access the similarity score matrix between document nodes and identify those high-order neighbors exhibiting substantial similarity to the anchor node, excluding them from negative sample selection. The presented neighbor hierarchical sifting loss significantly mitigates potential false negatives by accounting for the characteristics of neighbors across different hierarchical levels, thereby improving the contrastive learning process and enhancing the quality of learned node representations.

Contrastive loss: Based on the proposed negative selection strategy, we present a novel graph contrastive loss function neighbor hierarchical sifting loss (NHS). In this paper, node i in view r' is selected as the anchor node, its embedding is expressed as $u_i^{(r')}$, and its contrastive loss can be formulated as follows:

$$\mathcal{L}(u_i^{(r')}) = -log\frac{\xi_{inter}^{pos}}{\xi_{inter}^{pos} + \xi_{intra}^{neg} + \xi_{inter}^{neg}}, \tag{16}$$

the different terms in the above equation can be broken down into:

$$\xi_{inter}^{pos} = (\otimes\{e^{Dist(u_i^{(r')},u_i^{(r)})/\tau}\}|_{r=1}^{R}), \tag{17}$$

$$\xi_{intra}^{neg} = \sum_{v_j \subset D_i^{(r')}} (e^{Dist(u_i^{(r')},u_j^{(r')})/\tau}), \tag{18}$$

$$\xi_{inter}^{neg} = (\otimes\{\sum_{v_j \subset D_i^{(r)}} e^{Dist(u_i^{(r')},u_j^{(r)})/\tau}\}|_{r=1}^{R}), \tag{19}$$

where $R = \{title, keyword, event\}, r' \notin R$, $\otimes$ is the sum operator. $u_i^{(r)}$ is the representations of node i at the same position in view r. And $D_i^{(r)}$ is the negative sets of node i from view r. Specifically, the function $Dist(\cdot,\cdot)$ is instantiated as the cosine similarity measure, which quantifies the degree of similarity between two vector representations. The final contrastive loss NHS, defined as averaged over all nodes among the three views, is computed as follows:

$$\mathcal{L}_{NHS} = \frac{(\otimes\{\sum_{i=1}^{N}\mathcal{L}(u_i^{(r)})\}|_{r=1}^{R})}{q \cdot N}, \tag{20}$$

where $R = \{title, keyword, event\}$, $q = |R|$, $\otimes$ is the sum operator, and N is the number of node in contrastive view.

3.5. Label prediction

In the evaluation phase, we use the pre-trained RW-GCN and CGAN models to obtain text representations for the test data. For the text i, its final text representation is denoted as $\mathcal{T}_i$. Then, $\mathcal{T}_i$ will be input into a logistic regression classifier to obtain the classification results:

$$p_i = LRClassifier(\mathcal{T}_i). \tag{21}$$

where p_i denotes the predicted label of text i.

To sum up, the ConNHS can be summarized as Algorithm 1:

Algorithm 1 The overall process of ConNHS

```
Require: A text corpus $\boldsymbol{C}$, similarity threshold $\rho_t$, $\rho_k$, $\rho_e$, minimum association coefficient $\gamma_k$, $\gamma_e$.
$\boldsymbol{titles}$, $\boldsymbol{keywords}$, $\boldsymbol{events}$ = FeatureExtraction($\boldsymbol{C}$)
$\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{A}, \boldsymbol{R})$ = BuildGraph($\boldsymbol{titles}$, $\boldsymbol{keywords}$, $\boldsymbol{events}$, $\rho_t$, $\rho_k$, $\rho_e$, $\gamma_k$, $\gamma_e$)
$\hat{\boldsymbol{G}}_t$, $\hat{\boldsymbol{G}}_k$, $\hat{\boldsymbol{G}}_e$ = Separation($\boldsymbol{G}$)
for $t = 1$ to $T$ do
    $\boldsymbol{H}_t$, $\boldsymbol{H}_k$, $\boldsymbol{H}_e$ = $\boldsymbol{RW-GCN}(\hat{\boldsymbol{G}}_t, \hat{\boldsymbol{G}}_k, \hat{\boldsymbol{G}}_e)$;
    $\boldsymbol{H'} = \boldsymbol{CGAN}\ (\boldsymbol{H}_t, \boldsymbol{H}_k, \boldsymbol{H}_e)$;
    $\boldsymbol{U}_t$, $\boldsymbol{U}_k$, $\boldsymbol{U}_e$ = $\boldsymbol{Mapping}(\boldsymbol{H}_t, \boldsymbol{H}_k, \boldsymbol{H}_e)$;
    $\boldsymbol{score}$ = SimilarityAssessment($\boldsymbol{H'}$);
    $\boldsymbol{negatives}$ = NHS($\boldsymbol{V}, \boldsymbol{A}, \boldsymbol{score}$); /*Negative selection by NHS*/
    Compute contrastive loss $\mathcal{L}_{NHS}$ with the refined $\boldsymbol{negatives}$ via Eq. (16) and Eq. (20);
    Update parameters by applying gradient descent minimize $\mathcal{L}_{NHS}$.
end for
Get the text representations $\mathcal{T}$ via the pre-trained RW-GCN and CGAN.
Predict the labels of $\mathcal{T}$ via the logistic regression classifier.
return The predicted labels of each document node.
```

4. Experiments

In this section, we select four common text classification datasets and verify the effectiveness of our proposed method. Next, we will introduce the datasets and preprocessing, comparison methods, experimental settings, evaluation indicators, experimental results, and experimental result analysis.

4.1. Datasets and preprocessing

We select three news topic classification datasets (including two Chinese and one English news dataset) and a dataset in the medical field. A brief introduction to the dataset is as follows:

ThuCNews[3]: The ThuCNews corpus constitutes a news document collection derived through filtering the historical data from the Sina News RSS subscription channel spanning the period of 2005 to 2011, encompassing 14 news categories and comprising approximately 830,000 news articles. Considering the device factor and balancing the dataset, we randomly sample 5000 entries in each of the 14 categories.

SogouNews[4]: The SogouNews Corpus, furnished by SogouLabs, represents a news dataset encompassing SogouCA and SogouCS, comprising approximately 27,000 news items distributed across ten distinct categories. To attain a balanced distribution within the dataset, around 3000 samples were randomly sampled from each category, with the constraint that the character count of each sample exceeded 500.

Table 1
Summary statistics of the benchmark dataset.

	#Docs	#Train	#Test	#Classes	#Avg.Length
ThuCNews	84,000	67,200	16,800	14	539.75
SogouNews	30,000	24,000	6,000	10	502.4
20NG	18,846	15,076	3,770	20	221.26
Ohsumed	7,400	5,920	1,480	23	135.82

20NG[5]: The 20 News Corpus is an English text classification dataset containing newsgroup posts in 20 categories. There are 18,846 articles in total, with an average of about 1,000 articles per category.

Ohsumed[6]: The Ohsumed corpus is derived from the MEDLINE database, a bibliographic repository of significant medical literature curated by the National Library of Medicine. It encompasses 23 categories and a total of 7,400 articles. Given that each article is annotated with one or more tags, the highest-level tag is selected as the definitive label for the experimental setting.

Pre-processing: First, we filtered two Chinese news data sets according to the length of the text. The average length of the filtered news exceeded 500, which can verify the effectiveness of our proposed method in classifying longer texts. Secondly, noisy information unrelated to text category labels, such as the name of the author of the article and publication time, were removed from all datasets. Finally, Table 1 lists the summary statistics of the benchmark datasets.

4.2. Comparison of methods

In order to verify the effectiveness of our proposed method, we compared the four datasets mentioned above with the following eight state-of-the-art methods, which are:

PV-DBOW (Le & Mikolov, 2014): It is a paragraph vector model and ignores the word order in the text. Logistic regression is used as a classifier.

fastText (Joulin et al., 2017): The approach utilizes the mean of word/n-gram embeddings to represent document embeddings, subsequently feeding these aggregated vectors into a linear classifier for further analysis.

TextCNN (Kim, 2014): It is a type of traditional deep learning model and harnesses convolutional layers to autonomously and adaptively extract spatial hierarchies of features from the input data, thereby enabling the model to discern intricate patterns and relationships within the text.

RNN-Capsule (Wang et al., 2018): This model employs a capsule network-enhanced Recurrent Neural Network (RNN) for conducting sentiment analysis.

Bi-LSTM (Yao et al., 2019): A variant of the LSTM model is commonly used in text classification tasks.

Bert-large (Sun et al., 2024): It is a pre-trained language model based on the Transformer architecture. Based on the well-trained model, it is used for downstream text classification tasks after fine-tuning.

TextGCN (Yao et al., 2019): It is a model that uses graph convolutional neural networks for text classification. By building a graph structure and utilizing the representation learning capabilities of graph neural networks, it can effectively capture the semantic relationships between texts and improve the accuracy of text classification.

HAN (Wang et al., 2019): It proposes a novel dual-layer attention mechanism, encompassing node-level attention and semantic-level attention. Node-level attention is employed to quantify the salience of the relation between the centroid node and its heterogeneous neighboring

[3] http://thuctc.thunlp.org/

[4] https://huggingface.co/datasets/sogou_news

[5] http://qwone.com/~jason/20Newsgroups/

[6] https://disi.unitn.it/moschitti/corpora.htm

nodes, whereas semantic-level attention serves to ascertain the relative importance of distinct meta-paths.

RGCN (Schlichtkrull et al., 2018): It handles different types of nodes and relationship edges through relationship-specific graph convolution layers and node representation layers and obtains rich semantic information by iteratively updating node representations.

TGNCL (Li et al., 2023): It builds a graph for each document and develops a contrastive learning regularization to learn fine-grained word representations.

For all the aforementioned comparison methods, we adopted the recommended hyperparameter values for their configuration to ensure the optimal performance.

4.3. Experiment setting and evaluation criteria

In this section, we present the specific details of the experiment. Before conducting many experiments, the text will be preprocessed to remove irrelevant noise information. The second step is to extract the core feature information of the text (including keywords and events). Correspondingly, the title is a complete semantic sentence that can be obtained directly without special processing. For document nodes, which represent each text in this framework, the embedding representation encodes the text attributes using the LangChain framework, and the pre-trained embedding model BGE-M3 is used as the initial representation of the node. It is worth noting that our experimental results are the averages of 10 runs with different weight initializations. Additionally, the hyperparameter ρ_t is set to 0.7, while the default values for ρ_k and ρ_e are 0.6. The minimum association coefficient γ_e is set to 6 for long datasets and 3 for short datasets by default. Similarly, γ_k is set to 10 for long datasets and 6 for short datasets. The default value for the temperature parameter τ is 0.5. Unless otherwise specified, our proposed ConNHS adopts these default values in the following several experiments.

During the training process, if the training loss does not decrease for more than 50 consecutive epochs, the model is deemed to have reached convergence. Our method uses the Adam optimizer in the deep learning framework Pytorch. The training and testing processes of all datasets were completed on a computer equipped with Intel core i9-12900k CPU and Nvidia Geforce RTX3090.

We choose $Accuracy$, $Precision$, and $F1$ scores, common indicators in text classification tasks, to measure the effectiveness of our proposed method. $Accuracy$ represents the proportion of correctly classified samples to the total number of samples. $Precision$ indicates the proportion of correctly classified positive samples among all samples classified as positive. $F1$ takes into account precision and recall, making the evaluation more comprehensive. They can be formulated as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{22}$$

$$Precision = \frac{TP}{TP + FP} \tag{23}$$

$$F1 = \frac{2PR}{P + R} \tag{24}$$

where TP (True Positives) represents the number of samples correctly classified as category Y_i. FP (False Positives) refers to the number of samples from other categories incorrectly classified as Y_i. TN (True Negatives) indicates the number of samples from other categories correctly classified as not Y_i. FN (False Negatives) are the samples belonging to category Y_i but incorrectly classified into other categories. Additionally, R stands for Recall, which is the proportion of correctly predicted positive samples out of all actual positive samples.

4.4. Experiment results and analysis

4.4.1. Performance on text classification

Table 2 delineates the accuracy, precision, and F1 scores achieved by various methodologies across four benchmark datasets. Predominantly, the proposed ConNHS outperforms the baseline methods, showcasing its superior text classification prowess. The proposed ConNHS achieved accuracy improvements of 1.12, 0.30, 1.51, and 2.12 on the ThuCNews, SogouNews, 20NG, and Ohsumed datasets, respectively. We observed that the improvements of ConNHS on English datasets were more pronounced compared to the Chinese news datasets. This can be attributed to the fact that the baseline methods already achieved accuracy rates exceeding 90% on the Chinese news datasets. For Precision and F1 scores, ConNHS is likewise the most competitive method, consistently ranking among the top across all datasets. It is worth noting that RGCN demonstrated outstanding performance on the ThuCNews dataset, achieving the best Precision. Additionally, Bert-large and TGNLCL exhibited remarkable stability, with no significant performance fluctuations across multiple datasets. In contrast, the PV-DBOW model performed poorly in terms of precision and F1 score, lacking the competitiveness compared to deep learning models.

In a deeper analysis, we observe that there are also differences in classification capabilities between baseline methods. Firstly, the performance of deep learning-based baselines significantly surpassed that of word embedding models. Notably, Bert-large achieved performance competitive with GNN-based methods. This can be attributed to its pretraining on large-scale corpora and the bidirectional attention mechanism to understand each word in context, thereby possessing a strong semantic understanding capability. Besides, thanks to the fact that graph structures can construct relationships between texts, methods based on graph neural networks (including TextGCN, RGCN, and HAN) have achieved more outstanding classification accuracy than methods based on traditional deep learning. It is worth mentioning that TGNCL, a method based on graph contrastive learning, achieved highly competitive results but did not surpass our proposed ConNHS. This finding suggests that the graph augmentation adopted by TGNCL might, to some extent, disrupt critical semantic information in the text.

4.4.2. The effectiveness of the multi-relational text graph

To assess the effectiveness of our proposed multi-relational text graph (MTG) on text classification, we integrate MTG with other graph neural network models to observe the variations in text classification results.

The experimental results shown in Table 3 indicate that: by constructing semantic relationships based on core textual features, our multi-relational text graph effectively facilitates nodes learning richer semantic information from their diverse neighbors, thereby generating superior text representations. Upon leveraging our proposed multi-relational text graph, both the HAN and RGCN models exhibit an enhancement in classification accuracy performance. Notably, the HAN model has demonstrated a pronounced improvement in its performance across the two Chinese datasets, registering an accuracy gain exceeding 3%. The results manifest that despite the HAN model's capacity to adaptively acquire node representations via a dual attention mechanism, yielding excellent performance, the incorporation of multiple semantic relationships among documents offers additional perspectives, thereby enabling the HAN to capture more high-dimensional semantic information, consequently enhancing its learning capabilities. Conversely, the combination of the multi-relational text graph with the RGCN yields more substantial improvements in English datasets. While RGCN already showcases commendable performance on Chinese datasets, the integration of our proposed text graph still facilitates a certain level of advancement. By initiating the process with the extraction of core textual features and establishing multiple inter-text relationships, our approach effectively encourages models to assimilate and interpret high-dimensional semantic information. These experimental findings highlight the intrinsic value of multi-relational text graphs in enhancing text classification tasks.

Table 2
Test accuracy (%), P (%), and F1 score (%) for different models on two Chinese datasets and two English datasets.

Method	ThuCNews			SogouNews			20NG			Ohsumed		
	Acc	*P*	*F1*	*Acc*	*P*	*F1*	*Acc*	*P*	*F1*	*Acc*	*P*	*F1*
PV-DBOW	80.19	78.62	79.04	83.41	81.28	82.64	74.36	72.91	73.19	46.65	44.80	45.30
fastText	86.46	85.31	84.08	82.98	80.12	81.73	79.38	75.67	78.13	57.70	53.14	56.31
TextCNN	92.73	90.05	92.40	93.64	92.72	93.25	76.78	73.64	76.42	43.87	41.62	43.48
RNN-Capsule	85.52	84.25	83.21	86.43	85.32	85.92	73.18	72.49	73.02	49.37	46.98	49.10
Bi-LSTM	84.71	83.15	83.16	87.17	86.85	85.89	84.25	83.15	83.04	68.53	65.47	67.92
Bert-large	92.03	89.36	91.85	97.22	95.44	96.90	79.23	78.47	79.02	67.45	65.76	66.87
TextGCN	86.92	85.47	86.51	88.23	87.15	86.92	85.69	83.67	85.15	68.36	67.52	67.92
RGCN	94.74	**93.21**	92.33	93.62	91.09	92.16	78.72	77.06	77.45	67.51	64.78	65.90
HAN	86.17	84.67	83.08	89.06	87.36	88.52	79.86	78.26	77.30	68.20	65.14	67.51
TGNCL	94.10	90.12	93.27	96.37	94.25	95.18	85.92	84.86	85.13	67.82	66.47	66.03
ConNHS	**95.86**	93.14	**94.51**	**97.52**	**96.43**	**96.93**	**87.43**	**85.46**	**86.98**	**70.65**	**69.01**	**69.32**

Table 3
Test accuracy (%) and F1 score (%) for different models with multi-relational text graph.

Method	ThuCNews		SogouNews		20NG		Ohsumed	
	Acc	*F1*	*Acc*	*F1*	*Acc*	*F1*	*Acc*	*F1*
RGCN	94.74	92.33	93.62	92.16	78.72	77.45	67.51	65.90
RGCN_MTG	94.95(+0.21)	92.46(+0.13)	93.80(+0.18)	92.21(+0.05)	80.97(+2.25)	79.29(+1.84)	69.11(+1.60)	67.21(+1.31)
HAN	86.17	83.08	89.06	88.52	79.86	77.30	68.20	67.51
HAN_MTG	90.42(+4.25)	88.61(+5.53)	92.18(+3.12)	91.53(+3.01)	81.97(+2.11)	78.56(+1.26)	69.15(+0.95)	67.71(+0.20)
ConNHS	**95.86**	**94.51**	**97.52**	**96.93**	**87.43**	**86.98**	**70.65**	**69.32**

4.4.3. Text classification with few labels

The advantage of self-supervised GCL lies in its ability to train models using unlabeled data when labels are inaccessible or scarce. Our proposed ConNHS is designed for semi-supervised text classification tasks, requiring ground-true text labels during the testing phase. Therefore, we further test the performance of ConNHS in semi-supervised text classification under conditions of low label availability. We select different proportions of labeled data on the 20NG dataset to assess Bi-LSTM, TextGCN and the proposed ConNHS. To simulate scenarios of scarce labels, we set label rates at 1%, 2%, 5%, and 10%.

The results in Fig. 3 indicate that under conditions of low label rates, our proposed ConNHS exhibits superior classification performance. It is noteworthy that with only a sparse 1% of labeled text, our method still achieved an accuracy of 70.21%, while Bi-LSTM and TextGCN experienced a significant drop in performance. The reason behind this is that ConNHS effectively leverages large amounts of unlabeled data for training through self-supervised Graph Contrastive Learning (GCL). In contrast, Bi-LSTM and TextGCN do not incorporate any samples from the test set (unlabeled data) during the computation of training loss. The classification results with few labels indicate that, even with very sparse labeled text, our proposed method can be effectively applied to semi-supervised text classification tasks.

4.5. Ablation studies

To validate the effectiveness of our proposed contrastive loss NHS, this study conducted a series of ablation experiments on the ThuCnews, SogouNews, 20NG, and Ohsumed datasets. We design different experimental setups for the ablation study: employing the NT-Xent loss, removing the structure-guided signal, removing the attribute-guided signal, and utilizing the complete loss NHS. In these settings, NHS-na represents the removal of node attribute information as the guiding signal for negative sampling, leading to situations where high-order neighbors with high similarity in the graph might still be considered negative samples. NHS-gs denotes disregarding the graph homophily assumption, treating first-order neighbors of the anchor node as negative samples. Furthermore, NT-Xent, a well-established contrastive loss in graph contrastive learning, differs from NHS in that it considers both first-order neighbors and high-order similar neighbors of the anchor as negative samples. Through these ablation experiments, we aim to delve into how each component of the NHS specifically impacts model performance.

Table 4
Ablation experiment of NHS.

Method	ThuCNews		SogouNews		20NG		Ohsumed	
	Acc	*F1*	*Acc*	*F1*	*Acc*	*F1*	*Acc*	*F1*
NT-Xent	92.31	89.02	93.67	92.51	84.56	83.45	68.32	67.18
NHS	**95.86**	**94.51**	**97.52**	**96.93**	**87.43**	**86.98**	**70.65**	**69.32**
NHS-gs	94.38	93.34	95.56	95.08	85.63	85.34	69.56	68.41
NHS-na	95.04	94.82	96.21	95.69	86.27	85.97	69.94	68.84

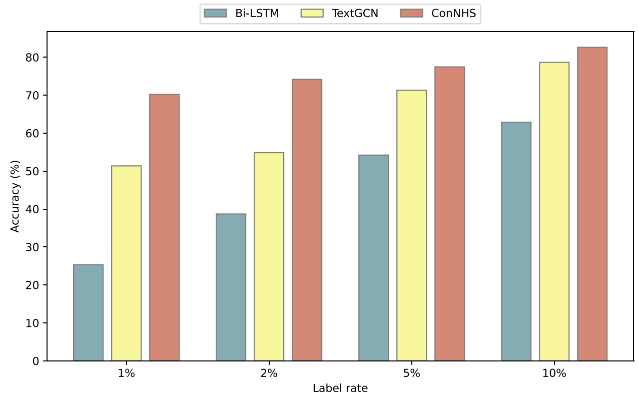

Fig. 3. The accuracy with few labels.

As shown in the ablation study results in Table 4, we observed that employing the NHS contrastive loss achieved the best performance on all four datasets. A decline in classification performance is noted when varying the contrastive loss of the ConNHS method, further highlighting the critical role of our proposed NHS contrastive loss in text classification tasks. Specifically, when switching to the NT-Xent contrastive loss, there is a decline in classification accuracy ranging between 3.33% to 5.55%. This result suggests that treating all remaining nodes in the graph as negatives inevitably increases the distance between similar document nodes in the embedding space, thereby reducing the accuracy of text classification. On the other hand, removing the graph structure information as the supervisory signal for negative sampling results in a decrease in accuracy ranging between 1.03% and 1.96%. Similarly, removing node attribute information also led to a certain

Table 5
Various hyperparameters.

Hyperparameter	Impact
ρ_t, ρ_e, ρ_k	These parameters determine whether there are similarities in titles, events, and keywords within the text. Their possible values range from [0.3, 0.9].
γ_e, γ_k	They individually dictate the degree of correlation in event relationships and keyword associations within the text. The value range for γ_e is [3, 8], while γ_k ranges from [5, 11].
τ	It regulates the model's sensitivity to variations in similarity. The adjustment range for τ is between 0.05 and 1.0.

degree of performance decline. Notably, the former scenario caused a more pronounced performance drop than the latter across all datasets. The underlying reason for this phenomenon is that the construction of the multi-relational text graph is based on the assumption of graph homophily, which posits that document nodes connected tend to have more similar core features and are more likely to belong to the same category. Therefore, excluding first-order neighbors of the anchor node from negative samples according to graph structure information aligns more closely with the objectives of graph contrastive learning. Overall, the results of the ablation experiments across different datasets conclusively demonstrate that our proposed NHS contrastive loss effectively mitigates false negative pairs and enhances the accuracy of text classification tasks.

4.6. *Parameters sensitivity*

In this section, we focus on exploring how various key parameters influence the performance of our method. It is worth pointing out that, inspired by the adjustment strategys of hyperparameters (Liu et al., 2024; Mo et al., 2022; Zhao & Song, 2023), we fix other hyperparameters as constants when investigating the impact of a particular hyperparameter on the performance of ConNHS. This allows for a direct observation of the impact of each hyperparameter on the model's performance. The details of the hyperparameters are illustrated in Table 5.

4.6.1. *The impact of similarity threshold*

Selecting an appropriate similarity threshold is vital for constructing a multi-relational text graph, as the establishment of the text graph is highly dependent on the degree of similarity between core features. To investigate the impact of changes in the similarity threshold on the performance of our method, we conducted a series of experiments and visualized the results for detailed analysis and reference. We performed independent experiments by sequentially varying the values of the title threshold, event threshold, and keyword threshold.

As demonstrated in Fig. 4, we observed a clear trend that the classification accuracy tends to increase as the ρ_t rises. However, it is noteworthy that once the ρ_t exceeds 0.7, the improvement in accuracy becomes more gradual. In terms of event feature analysis, increasing the ρ_e indeed effectively boosts accuracy, a trend that continues until the threshold reaches 0.6. Beyond this point, further increases in the ρ_e lead to a decrease in accuracy. This indicates that overly high similarity thresholds might reduce the connections between similar texts, weakening the model's ability to learn textual information under event relations. Additionally, we found that increasing the ρ_k also enhances classification accuracy, but this trend reverses when the ρ_k exceeds 0.6. For all core features, the model performs worse when the similarity threshold is too low. This may be because a low threshold creates redundant edges between text nodes. An appropriate threshold, on the other hand, establishes more reliable connections, thereby improving the node representations learned by the graph neural network. Experimental results indicate that the optimal ρ_t is 0.7. While ρ_e is 0.6, ConNHS achieved the best performance across all four datasets with these settings. For ρ_k, the optimal threshold range is between 0.6 and 0.7.

4.6.2. *The impact of minimum association coefficient*

Texts that share semantically similar events or keywords tend to belong to the same domain. Therefore, we evaluate the impact of different minimum association coefficients γ_e and γ_k on the performance of the proposed ConNHS in text classification tasks.

As shown in Fig. 5, we observe that the accuracy of text classification increases with the rise of γ_e and γ_k. Overall, compared to the Chinese dataset, the two English datasets, which have shorter average lengths, achieve optimal results more quickly. Specifically, when the value of γ_e is 3, ConNHS achieves the highest classification accuracy on the 20NG and Ohsumed datasets. In contrast, when γ_e is set to 6 and 7, the corresponding accuracies for ThuCnews and SogouNews are better. For the minimum association threshold γ_k, when its value is 6, the 20NG and Ohsumed datasets achieve the best results. In contrast, ThuCnews and SogouNews achieve the highest classification accuracy when γ_k is set to 9 and 10, respectively. It is worth noting that for shorter datasets, after reaching the highest accuracy, further increasing the values of γ_e and γ_k leads to a significant decline in performance. The experimental results reveal a trend that for longer datasets, the optimal values of γ_e and γ_k tend to be higher than those for shorter datasets. This is because short texts have limited feature information, and setting the minimum association threshold too high may cause many potential semantic connections to be overlooked, thereby reducing effective links between texts.

4.6.3. *The impact of temperature hyperparameter*

The temperature parameter plays a pivotal role in graph contrastive learning as a fundamental hyperparameter that modulates the distribution of similarity scores within the contrastive loss function. To analyze the impact of the temperature parameter on classification accuracy, we conduct validation on the ThuCnews and 20NG datasets. As shown in Fig. 6, the results indicate that a too-low temperature parameter leads to suboptimal classification outcomes. As the value of τ increases, the model classification capability improves, and our method achieves the best results on both ThuCnews and 20NG when τ is approximately 0.5. It is noteworthy that a value that is too high for the temperature parameter can also lead to a decline in performance. The experimental results suggest that the value of τ may require fine-tuning for different datasets to achieve optimal performance. In general, we recommend starting with a value of 0.5 and conducting a thorough parameter search within the range of 0.4 to 0.7.

5. Conclusion

In this paper, the ConNHS method we propose demonstrates competitive performance in semi-supervised text classification tasks. Firstly, inspired by the logic humans use to categorize texts, we constructed a multi-relational text graph. Subsequently, we introduced RW-GCN and CGAN for intra-graph and inter-graph propagation, respectively. RW-GCN leverages edge features to capture varying correlations between nodes, while CGAN learns the differences in inter-graph features and integrates document node representations. Additionally, we introduced the neighbor hierarchical sifting loss to optimize the negative selection process, effectively mitigating the issue of false negatives. Extensive experiments conducted on multiple datasets demonstrate that our method achieves superior results across various evaluation metrics compared to existing approaches. It is worth noting that the multi-relational graphs we constructed inevitably contain some noisy edges, which may mislead the learning process of the graph neural networks. In future work, we will explore denoising techniques in multi-relational text graphs to further optimize the node aggregation process and enhance model performance. Meanwhile, we will continue to investigate graph contrastive learning, with a particular focus on optimizing the negative sample selection process.

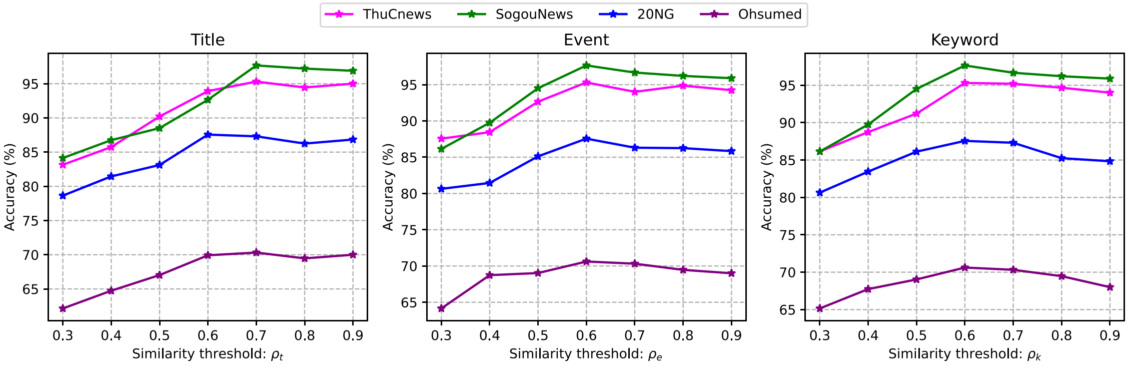

Fig. 4. The ConNHS performance under different similarity threshold of core features.

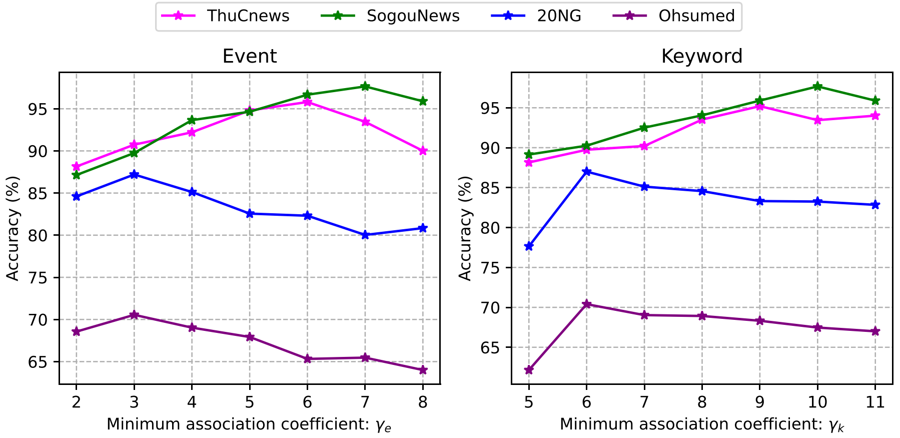

Fig. 5. The ConNHS performance under different minimum association coefficient.

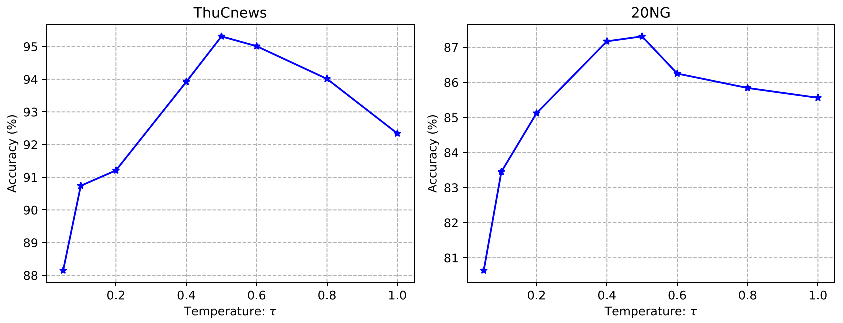

Fig. 6. The ConNHS performance under different temperature.

CRediT authorship contribution statement

Wei Ai: Supervision, Investigation, Writing – review & editing. **Jianbin Li:** Conceptualization, Methodology, Investigation, Data curation, Writing – original draft. **Ze Wang:** Supervision, Writing – review & editing. **Yingying Wei:** Supervision, Investigation, Review. **Tao Meng:** Supervision, Investigation, Writing – review & editing. **Keqin Li:** Supervision, Investigation, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by National Natural Science Foundation of China (Grant No. 69189338), Excellent Young Scholars of Hunan Province of China (Grant No. 22B0275), and Changsha Natural Science Foundation (Grant No. kq2202294).

Data availability

Data will be made available on request.

References

Chang, W. C., Yu, H. F., Zhong, K., Yang, Y., & Dhillon, I. S. (2020). Taming pretrained transformers for extreme multi-label text classification. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 3163–3171).

Chen, J., Xiao, S., Zhang, P., Luo, K., Lian, D., & Liu, Z. (2024). Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. arXiv preprint arXiv:2402.03216.

Dieng, A. B., Wang, C., Gao, J., & Paisley, J. (2016). TopicRNN: A recurrent neural network with long-range semantic dependency. arXiv preprint arXiv:1611.01702.

Hassani, K., & Khasahmadi, A. H. (2020). Contrastive multi-view representation learning on graphs. In *International conference on machine learning* (pp. 4116–4126). PMLR.

Huang, L., Ma, D., Li, S., Zhang, X., & Wang, H. (2019). Text level graph neural network for text classification. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing* (pp. 3444–3450).

Joulin, A., Grave, É., Bojanowski, P., & Mikolov, T. (2017). Bag of tricks for efficient text classification. In *Proceedings of the 15th conference of the European chapter of the Association for Computational Linguistics: volume 2, short papers* (pp. 427–431).

Kenton, J. D. M. W. C., & Toutanova, L. K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT* (pp. 4171–4186).

Kim, Y. (2014). Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882.

Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. In *International conference on learning representations* (pp. 1–10).

Lai, S., Xu, L., Liu, K., & Zhao, J. (2015). Recurrent convolutional neural networks for text classification. In *Proceedings of the AAAI conference on artificial intelligence, vol. 29, no. 1* (pp. 1–10).

Lan, G., Hu, M., Li, Y., & Zhang, Y. (2023). Contrastive knowledge integrated graph neural networks for Chinese medical text classification. *Engineering Applications of Artificial Intelligence*, *122*, Article 106057.

Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. In *International conference on machine learning* (pp. 1188–1196). PMLR.

Lei, F., Liu, X., Li, Z., Dai, Q., & Wang, S. (2021). Multihop neighbor information fusion graph convolutional network for text classification. *Mathematical Problems in Engineering*, *2021*, 1–9.

Li, C., Peng, X., Peng, H., Li, J., & Wang, L. (2021). TextGTL: Graph-based transductive learning for semi-supervised text classification via structure-sensitive interpolation. In *IJCAI* (pp. 2680–2686).

Li, X., Wang, B., Wang, Y., & Wang, M. (2023). Graph-based text classification by contrastive learning with text-level graph augmentation. *ACM Transactions on Knowledge Discovery from Data*.

Lin, Y., Meng, Y., Sun, X., Han, Q., Kuang, K., Li, J., & Wu, F. (2021). BertGCN: Transductive text classification by combining GNN and BERT. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021* (pp. 1456–1462).

Linmei, H., Yang, T., Shi, C., Ji, H., & Li, X. (2019). Heterogeneous graph attention networks for semi-supervised short text classification. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing* (pp. 4821–4830).

Liu, X., Ma, K., Wei, Q., Ji, K., Yang, B., & Abraham, A. (2024). G-HFIN: graph-based hierarchical feature integration network for propaganda detection of we-media news articles. *Engineering Applications of Artificial Intelligence*, *132*, Article 107922.

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

Liu, X., You, X., Zhang, X., Wu, J., & Lv, P. (2020). Tensor graph convolutional networks for text classification. In *Proceedings of the AAAI conference on artificial intelligence, vol. 34, no. 05* (pp. 8409–8416).

Miao, R., Yang, Y., Ma, Y., Juan, X., Xue, H., Tang, J., Wang, Y., & Wang, X. (2022). Negative samples selecting strategy for graph contrastive learning. *Information Sciences*, *613*, 667–681.

Mo, Y., Peng, L., Xu, J., Shi, X., & Zhu, X. (2022). Simple unsupervised graph representation learning. In *Proceedings of the AAAI conference on artificial intelligence, vol. 36, no. 7* (pp. 7797–7805).

Piao, Y., Lee, S., Lee, D., & Kim, S. (2022). Sparse structure learning via graph neural networks for inductive document classification. In *Proceedings of the AAAI conference on artificial intelligence, vol. 36, no. 10* (pp. 11165–11173).

Schlichtkrull, M., Kipf, T. N., Bloem, P., Van Den Berg, R., Titov, I., & Welling, M. (2018). Modeling relational data with graph convolutional networks. In *The semantic web: 15th international conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, proceedings 15* (pp. 593–607). Springer.

Shen, X., Sun, D., Pan, S., Zhou, X., & Yang, L. T. (2023). Neighbor contrastive learning on learnable graph augmentation. In *Proceedings of the AAAI conference on artificial intelligence, vol. 37, no. 8* (pp. 9782–9791).

Shi, S., Hu, K., Xie, J., Guo, Y., & Wu, H. (2024). Robust scientific text classification using prompt tuning based on data augmentation with L2 regularization. *Information Processing & Management*, *61*(1), Article 103531.

Sun, G., Cheng, Y., Zhang, Z., Tong, X., & Chai, T. (2024). Text classification with improved word embedding and adaptive segmentation. *Expert Systems with Applications*, *238*, Article 121852.

Sun, Z., Harit, A., Cristea, A. I., Yu, J., Shi, L., & Al Moubayed, N. (2022). Contrastive learning with heterogeneous graph attention networks on short text classification. In *2022 international joint conference on neural networks* (pp. 1–6). IEEE.

Tai, K. S., Socher, R., & Manning, C. D. (2015). Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd annual meeting of the Association for Computational Linguistics and the 7th international joint conference on natural language processing (volume 1: long papers)* (pp. 1556–1566).

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). Graph attention networks. In *International conference on learning representations* (pp. 1–10).

Wang, X., Ji, H., Shi, C., Wang, B., Ye, Y., Cui, P., & Yu, P. S. (2019). Heterogeneous graph attention network. In *The world wide web conference* (pp. 2022–2032).

Wang, Y., Sun, A., Han, J., Liu, Y., & Zhu, X. (2018). Sentiment analysis by capsules. In *Proceedings of the 2018 world wide web conference* (pp. 1165–1174).

Wang, Y., Wang, C., Zhan, J., Ma, W., & Jiang, Y. (2023). Text FCG: Fusing contextual information via graph learning for text classification. *Expert Systems with Applications*, Article 119658.

Xia, J., Wu, L., Chen, J., Hu, B., & Li, S. Z. (2022). Simgrace: A simple framework for graph contrastive learning without data augmentation. In *Proceedings of the ACM web conference 2022* (pp. 1070–1079).

Xu, D., Cheng, W., Luo, D., Chen, H., & Zhang, X. (2021). Infogcl: Information-aware graph contrastive learning. *Advances in Neural Information Processing Systems*, *34*, 30414–30425.

Yang, H., Chen, H., Pan, S., Li, L., Yu, P. S., & Xu, G. (2022). Dual space graph contrastive learning. In *Proceedings of the ACM web conference 2022* (pp. 1238–1247).

Yang, Y., Miao, R., Wang, Y., & Wang, X. (2022). Contrastive graph convolutional networks with adaptive augmentation for text classification. *Information Processing & Management*, *59*(4), Article 102946.

Yao, L., Mao, C., & Luo, Y. (2019). Graph convolutional networks for text classification. In *Proceedings of the AAAI conference on artificial intelligence, vol. 33, no. 01* (pp. 7370–7377).

You, Y., Chen, T., Sui, Y., Chen, T., Wang, Z., & Shen, Y. (2020). Graph contrastive learning with augmentations. *Advances in Neural Information Processing Systems*, *33*, 5812–5823.

Zhang, N., Deng, S., Ye, H., Zhang, W., & Chen, H. (2022). Robust triple extraction with cascade bidirectional capsule network. *Expert Systems with Applications*, *187*, Article 115806.

Zhang, S., Wang, L., Sun, K., & Xiao, X. (2020). A practical Chinese dependency parser based on a large-scale dataset. arXiv preprint arXiv:2009.00901.

Zhang, H., & Zhang, J. (2020). Text graph transformer for document classification. In *Conference on empirical methods in natural language processing* (pp. 1–9).

Zhang, Y., Zhang, Y., Qi, P., Manning, C. D., & Langlotz, C. P. (2021). Biomedical and clinical english model packages for the stanza python NLP library. *Journal of the American Medical Informatics Association*, *28*(9), 1892–1899.

Zhao, Y., & Song, X. (2023). TextGCL: Graph contrastive learning for transductive text classification. In *2023 International joint conference on neural networks* (pp. 1–8). IEEE.

Zhu, Y., Xu, Y., Yu, F., Liu, Q., Wu, S., & Wang, L. (2021). Graph contrastive learning with adaptive augmentation. In *Proceedings of the web conference 2021* (pp. 2069–2080).