IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 3 February 2023, accepted 21 February 2023, date of publication 27 February 2023, date of current version 2 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3250109

RESEARCH ARTICLE

Enhancing Text Classification by Graph Neural Networks With Multi-Granular Topic-Aware Graph

YONGCHUN GU[1], YI WANG[2], HENG-RU ZHANG[3], JIAO WU[4], (Member, IEEE), AND XINGQUAN GU[5]

[1]School of Mathematics, Sichuan University of Arts and Sciences, Dazhou 635000, China
[2]Key Laboratory of Intelligent Education Technology and Application of Zhejiang Province, Zhejiang Normal University, Jinhua 321004, China
[3]School of Computer Science, Southwest Petroleum University, Chengdu 610500, China
[4]College of Science, China Jiliang University, Hangzhou 310018, China
[5]College of Standardization, China Jiliang University, Hangzhou 310018, China

Corresponding author: Yi Wang (kelly_sylvia@163.com)

This work was supported in part by the Chinese National Natural Science Foundation under Grant 11701540, in part by the China State Administration for Market Regulation Technical Support Special Project 2021YJ005, in part by the Open Research Fund of College of Teacher Education, in part by the Zhejiang Normal University under Grant jykf22004, in part by the Multi-Dimensional Data Perception and Intelligent Information Processing Laboratory Open Fund Project DWSJ2213, and in part by the Sichuan University of Arts and Sciences Research Initiation Fund under Grant 2022QD68.

ABSTRACT Text classification based on graph neural networks (GNNs) has been widely studied by virtue of its potential to capture complex and across-granularity relations among texts of different types from learning on a text graph. Existing methods typically construct text graphs based on words-documents to capture relevant intra-class document representations among the same documents via words-words and words-documents propagation. However, a natural problem is that polysemy words in documents may become an information medium between documents of different categories, promoting heterophily information propagation. The performance of text classification will be somewhat constrained by this issue. This paper proposes a novel text classification method based on GNN from multi-granular topic-aware perspective, referred to as Text-MGNN. Specifically, topic nodes are introduced to build a triple node set of ''word, document, topic,'' and multi-granularity relations are modeled on a text graph for this triple node set. The introduction of topic nodes has three significant advantages. The first is to strengthen the propagation of topics, words, and documents. The second is to enhance class-aware representation learning. The final is to mitigate the effect of heterophily information caused by polysemy words. Extensive experiments are conducted on three real-world datasets. Results validate that our proposed method outperforms 11 baselines methods.

INDEX TERMS Graph neural networks, text classification, text graph construction.

I. INTRODUCTION

Text classification is one of the most fundamental task of Natural Language Processing (NLP), which has achieved good performance in the fields of emotion analysis [1], [2], information retrieval [3] and spam detection [4]. The core design of models for text classification is on the study of texts representation learning. The learned representation can be used to implement the accurate text classification if these learned representation of texts are class distinguishable [5].

Traditional text representation methods, such as vector space model [6], require artificially designed features. Such traditional text representation methods have attracted widespread attention due to their simple operation, but the text representations obtained also suffer from sparse, high-dimensional and other shortcomings. In recent years, with

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

the rapid development of deep learning in the field of natural language processing [7], a large number of text feature extraction models based on classical neural network, such as convolutional neural network (CNN) [8], [9] and recurrent neural network (RNN) [10], [11], have emerged. These two methods learn text feature representations in an end-to-end manner and solve the problem of sparse, high-dimensional text representations, however, these methods have some limitation of learning global word co-occurrences in corpora with discontinuous and long-range semantics.

Recently, text classification methods based on graph neural network (GNN) have break through the above mentioned limitation. Graph neural network can capture the global information of nodes in the graph by message passing mechanism over graphs, which has great successes in a lot of fields [12], [13], [14]. The important challedge with this type of approaches is to build suitable text graphs. TextGCN [15], as a classical GNN model for text classification, regards documents and words as nodes of graph and converts the text dataset into a large heterogeneous graph with two-granular texts. Then use graph neural network to classify documents nodes. Consistent representation learning for intra-class documents and distinguishable representation learning for inter-class documents both depend on words representations learning and the information propagation of words among documents. However, same word may have different semantics in different topics, such as, ''apple'' refers to the Apple Company in the field of technology, and refers to fruit in the field of food. Such words will be connected with multiple categories of documents and then prompt the propagation among heterophily information of documents to affect the performance of downstream tasks such as text classification. Figure 1 (a) counts the proportion of such polysemy words in documents in the real dataset, taking the R8 dataset as an example, the total number of categories of documents in this dataset is 8. From the statistical results in the figure, it can be seen that each category of documents contains polysemy words, which shows that polysemy words are common. In addition, Figure 1 (b) further show that the problem of heterophily information flow among different types of documents will be caused by these polysemy words. w_1 is a polysemy word, which is an information medium between d_1 and d_2, including double information. Under the action of the message passing mechanism, w_1 not only receives the information of d_1 and d_2 at the same time, but also propagates the received information back to d_1 and d_2 respectively, thus forming the heterophily information propagation between d_1 and d_2, which is an urgent problem to be solved.

To alleviate the above problem, we propose a GNN method for text classification from a multi-granular topic-aware perspective, referred to as Text-MGNN. Text-MGNN introduces topic nodes in the text graph to enhance the relations among words, documents and category-attribute features. Specifically, we construct multi-granular topic-aware graphs (MGTA graphs) as in Figure 1 (b), which add topic-aware information for all text (i.e., documents and words) in training set to enhance class-aware representation learning by adding the information flow with topics knowledge, accordingly improving the performance of text classification. In addition, we define multi-granular relations over the MGTA graphs, which include word-word relations to help learn the underlying information with contextual semantics, word-document relations to help further capture ego information among documents with a certain difference, the introduced topic-word relations and topic-document relations can enhance the upper information learning that is with respect to more class-aware representation for documents.

Overall, focusing on the effect of polysemy words on the text graphs, we present a novel text graph construction methods from multi-granular topic-aware perspective. This constructed text graph can enhance text classification. In particular, our contributions are three-fold:

- We introduce a novel GNN-based method for text classification named Text-MGNN to enhance text classification by alleviating the heterophily information propagation among inter-class documents caused by polysemy words.
- We learn multi-granular relations among a triple nodes ''words-topic-document'' to promote the class-aware representation learning for documents by defining four types of relations computing methods respectively.
- Text-MGNN is validated by conducting comprehensive experiments on several public datasets, outperforming the extensive range of representative baselines.

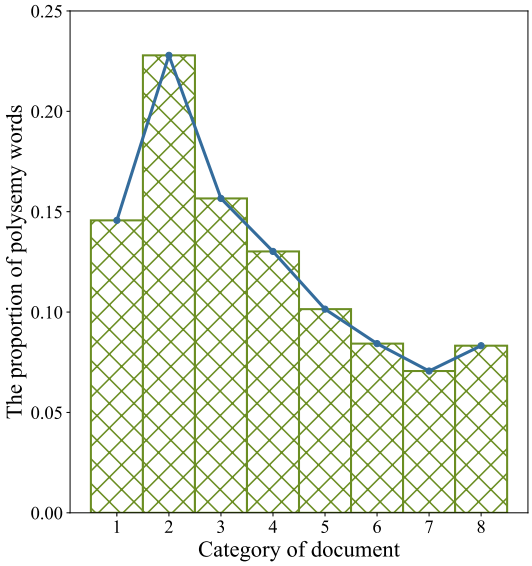
The remainder of this paper is organized as follows. Section II briefly overviews three types of popular methods for text classification. Section III gives the description of classical graphs and multi-granular topic-aware graphs. Section IV presents the proposed Text-MGNN method. Experimental results are validated and analyzed in Section V. Conclusions are drawn in Section VI.

II. RELATED WORK

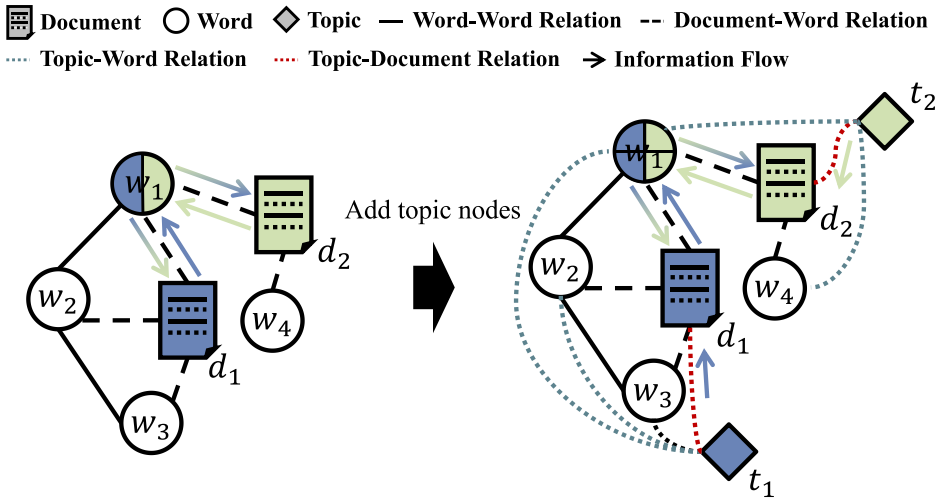
Current text classification methods can been classified into three categories, such as word embedding based methods [16], classical neural networks based methods [17] and graph neural networks based methods [18].

A. WORD EMBEDDING BASED METHODS

Word embedding based methods are usually used to implement text classification in the early researches. The basic idea is to use feature engineering methods to learn text representation, and then train a classifier to predict the category of the text. The common classifiers include support vector machine (SVM) [19], naive bayesian (NB) [20], k-nearest neighbors (KNN) [21], etc.. The core design of these methods focus on feature engineering, including two operations, i.e., feature extraction and feature selection. For example, Joachims [22] consider a bag-of-words model to map the text into a fixed-length vector at first, and then



(a)



(b)

FIGURE 1. (a) A schematic diagram of the proportion count of polysemy words for each category of document on R8 dataset. (b) A schematic diagram of information flow among documents over two text graphs with and without topic information.

achieve feature dimensionality reduction by the information gain criterion. Finally, using SVM classifier iteratively trains the processed feature, thereby achieves a good classification effect. Besides the bag-of-words model, the n-grams model proposed in [23] is also often used for feature extraction. However, the word embedding based methods have two main sides of drawbacks. One is the computational memory limitation under a huge number of parameters in large corpus. Another is the sparsity of features obtained by such feature selection methods.

B. CLASSICAL NEURAL NETWORKS BASED METHODS

Compared with word embedding based methods, classical neural networks based methods can handle large corpus problem. Kim [9] used CNN to extract sentence features and achieved good results in sentence classification. Zhang et al. [24] consider text as a kind of raw signal at character level and applied CNN to extract text feature, achieving promising results. In addition, text is sequence data, while RNN, LSTM and their variants are often used to process this type of data. Lai et al. [10] applied RNN to capture the context information of words, and used the max pooling to learn the key elements in the text, and achieved good results on text classification. Sinha et al. [25] used the BiLSTM to encode the word into the representation based on the word context information, and then used the perceptron to predict the category of the text. Although methods based on classical neural networks can effectively improve the performance of text classification, these methods do not consider the global co-occurrence information of words, and

like word embedding based methods, they can only process data in Euclidean space [26].

C. GRAPH NEURAL NETWORKS BASED METHODS

Text is non-Euclidean structured data, word embedding based and classical neural networks based methods cannot directly learn such Non-Euclidean data, representation of which should be transformed into Euclidean space for further processing. Graph neural networks (GNNs) based methods can process this type of data. TextGCN [15] is the first research to apply graph convolutional networks to text classification tasks. This model regards words and documents as nodes and constructs an undirected weighted heterogeneous graph for the entire corpus. Then employing graph convolutional network (GCN) learns word embedding and documents embedding to implement text classification. In addition, a classical GNN model, SGC [27], can be directly used to implement text classification, which is effectively validated on several benchmark text classification datasets in the experiment. Li et al. [28] focused on the problem of spam detection and designed a model based GNNs, named as GCN-based Anti-Spam (GAS) model. This model integrated heterogeneous and homogeneous graphs to capture the local and global contexts of reviews. Although these methods bring better performance for text classification, however, they only construct text graph with keywords and documents, which ignores the problem of heterophily information progragetion among heterophily documents faced by polysemous words under the GNN-based message propagation mechanism. Our model effectively alleviates the influence of heterophily information by adding topic information during graph construction.

III. PRELIMINARIES

Assumed that there is a document corpus $\Omega = \{d_1, d_2, \ldots, d_{N_\Omega}\}$, a label set $\mathcal{Y} = \{y_1, y_2, \ldots, y_C\}$, and a vocabulary set $\mathcal{W} = \{w_1, w_2, \ldots, w_{N_\mathcal{W}}\}$, where N_Ω, $N_\mathcal{W}$ respectively denote the number of documents and words, and C is the total number of label class. In addition, the label class can be converted to representations with topic information via one-hot encoding [29], denoting as $\mathcal{T} = \{t_1, t_2, \ldots, t_C\}$. The purpose of text classification is to find the most suitable label y_{d_i} for unseen documents $d_i = \{w_{i_1}, w_{i_2}, \ldots, w_{i_{|d_i|}}\}$ based on learned inductive representations of documents of known categories. Recently, GNNs are widely concerned among a variety of applications, for their strong learning ability of unstructured data. Text data is one of the classical unstructured data. The introduction of GNNs will bring a positive performance for learning such types of data. Compared with traditional text classification, text classification based on graphs should convert the text data into graph-structured data at first, then implement the text classification by feeding the text graph into a GNNs model based on node classification. A challenging problem is how to construct a reasonable text graph. Consider text data is usually mutiple-granular, thus formally, text graph can

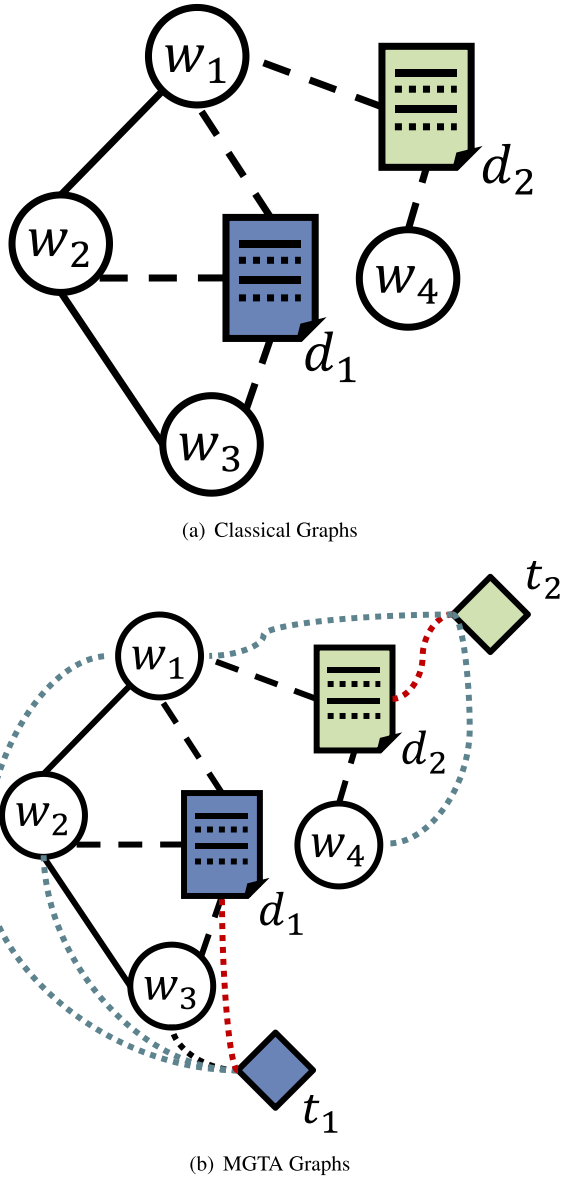


FIGURE 2. The schematic diagram of the difference between classical graphs and MGTA graphs.

be modeled as $\mathcal{G} = \left(\bigcup_{i=1}^{N_n} \mathcal{V}_i, \bigcup_{i=1}^{N_e} \mathcal{E}_i, \mathbf{X}\right)$, where $\mathcal{V}_i$ is the set of i-th types of nodes set, and $\mathcal{E}_i$ is the set of edges set that denotes i-th types of relations set connection between two nodes, N_n and N_e are the total number of categories for nodes and edges, respectively. $\mathbf{X} \in \mathbb{R}^{\sum_{i=1}^{N_n} |\mathcal{V}_i| \times d}$ denotes a feature matrix of all nodes, where d is the intial feature dimension. In general, a text graph at least contains document nodes, i.e., $N_n \geq 1$, and satisfy $N_e \geq 1$, otherwise there is no graph structure. Furthermore, if $N_n \geq 2$, we called it as heterogeneous graph, and based on the condition of $N_n \geq 2$, if $N_e \geq 2$, the text graph can be referred to as multi-granular graphs. The classical graphs, like in TextGCN, N_n is set 2, including document nodes and word nodes. In this paper, we further designs a multi-granular topic-aware graphs construction method by introducing topic nodes to

enhance the class-aware information of document nodes. The difference between classical graphs and multi-granular topic-aware graphs (MGTA graphs) is illustrated in Figure 2. For readability, we give a definition of text graph in Definition 1, as well as all the above-mentioned symbols and the notations using in the following paper are listed in Table 7 in the appendix.

*Definition 1 (**MGTA Graphs**):* A text graph can be given as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$, where $\mathcal{V} = \bigcup_{i=1}^{N_n} \mathcal{V}_i$ and $\mathcal{V}_i$ is the set of i-th types of nodes set, $\mathcal{E} = \bigcup_{i=1}^{N_e} \mathcal{E}_i$ and $\mathcal{E}_i$ is the set of edges set that denotes i-th type of relations set connection between two nodes. In addition, $\mathbf{X} \in \mathbb{R}^{\sum_{i=1}^{N_n} |\mathcal{V}_i| \times d}$ is the feature matrix with d dimension. When convenient, let $\mathcal{G} = (\mathcal{V}, \mathbf{A}, \mathbf{X})$ is equivalently represented as a graph, where $\mathbf{A} \in \mathbb{Z}^{|\mathcal{V}| \times |\mathcal{V}|}$ is an adjacency matrix representation of the edges in $\mathcal{E}$. In the common, classical graphs satisfies $N_n = N_e = 2$, while $N_n \geq 2$ and $N_e \geq 2$ denotes the multi-granular graph; in particular, the defined MGTA graphs in this paper satisfies $N_n = 3$ and $N_e = 4$.

IV. TEXT-MGNN: MULTI-GRANULAR TOPIC-AWARE GRAPH LEARNING FOR TEXT CLASSIFICATION

This section overviews Text-MGNN, which is a novel graph learning method for text classification based on multi-granular topic-aware graph, as illustrated in Figure 3. The core of Text-MGNN is on the text graph construction. Recall that TextGCN constructs the text graph by word co-occurrence and document word relations and then enhances the effective feature representation extraction for the document via GNNs. However, the representations over documents and words are linearly independent within and across categories according to the common text representation method. The natural indistinguishability over text representations on graph is easy to limit the learning of more distinguishable semantic representations for documents. We focus on the multi-granularity information of documents, i.e., a triple relation ''words-topic-document'', for enhancing the similarity representations between text (including words and documents) and topic, accordingly strengthen the more unbiased representations of documents. Specifically, we present a multi-granular topic-aware graph construction method (MGTA) that introduce topic informations to the document and its words under the known classes, thereby enhancing word co-occurrence and document word relations. Then, we further employ GNNs over constructed multi-granularity text graph to learn the multi-granularity representation for document. Under a more strong representation of document, the text classification performance will be increased.

In the following, we introduce the MGTA and Text-MGNN, and the overall objective.

A. MULTI-GRANULARITY TOPIC-AWARE GRAPH CONSTRUCTION

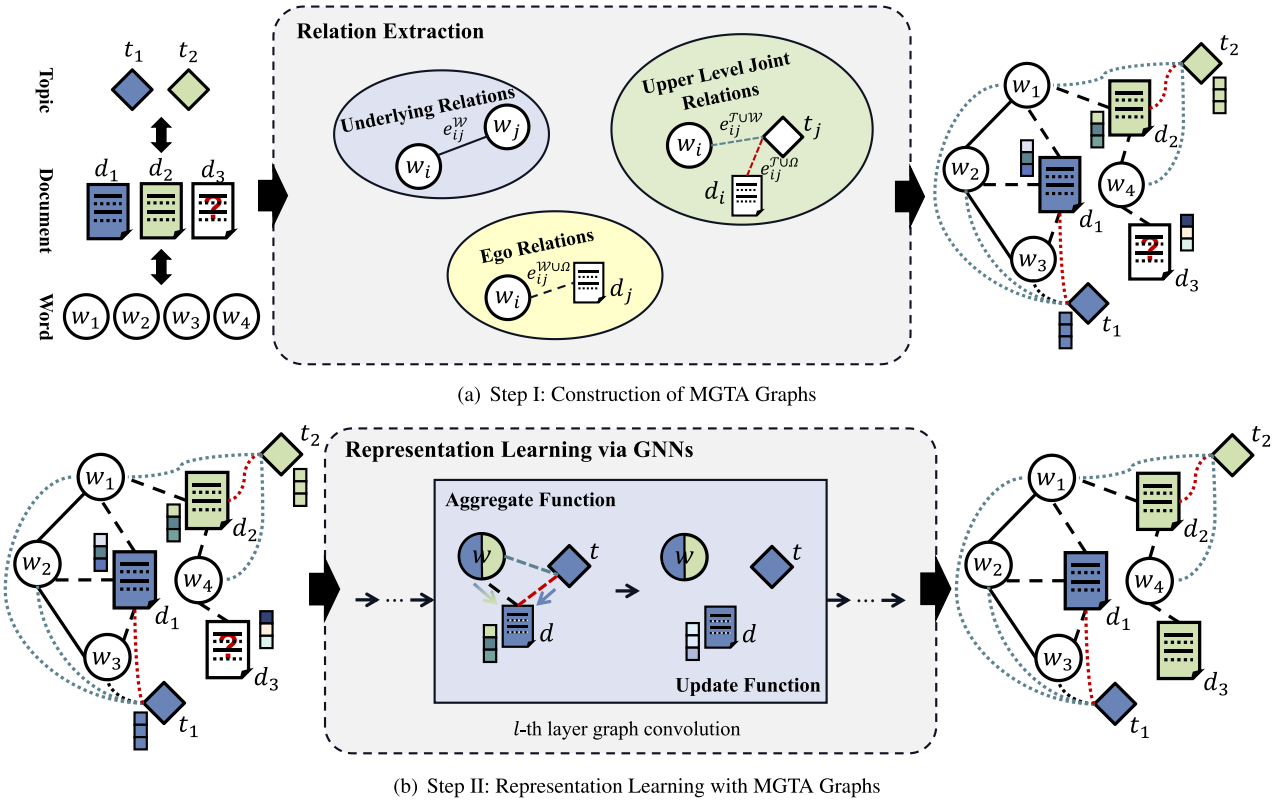
Text classification usually aims at predicting the attributive topic of the document. Preliminarily, text graph construction is the foundational work for text classification using GNNs, we bulid a text graph from the perspective of multi-granularity over text semantics, which is shown in Figure 3 (a). In common, a topic contains multiple documents, while a document includes a number of sentences, where a sentence is composed of words one by one. For instance, a sentence ''To be, or not to be: that is the question'' can be split into eight words, i.e., ''to'', ''be'', ''or'', ''not'', ''that'', ''is'', ''the'' and ''question''. The sentence is from the document ''Hamlet'', which is a famous tragic novel. To this end, for this document, its information is composed of a triple relation. One is the upper level information, such as some abstract information, i.e., attributive topic that is ''tragic novel''. The second is middle level information, the ego information of documents. The third is the underlying information, which is the integrated information over all words, which belongs to the document. We extract this triple relation as ''words-topic-document''. It should be noted that in order to strengthen the constraints of topics to document of known category, we not only established the relation between ''topic'' and ''document'', but also ''topic'' and ''words'', due to some words are with unobservable connection with the topic, e.g., words ''question'' and topic ''tragic novel''. The triple relation extraction for document forms a bottom-up information joint. Define text graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{X}\}$, where $\mathcal{V} = \Omega \bigcup \mathcal{W} \bigcup \mathcal{T}$ and $\mathcal{E} = \mathcal{E}_{\{w,w\}} \bigcup \mathcal{E}_{\{w,d\}} \bigcup \mathcal{E}_{\{t,w\}} \bigcup \mathcal{E}_{\{t,d\}}$. $\mathcal{E}_{\{w,w\}}$ denotes the relations among words, which is used to model the underlying relations. $\mathcal{E}_{\{w,d\}}$ is the across relations between ego information and underlying information, i.e., ''words-document'' relations. $\mathcal{E}_{\{t,w\}}$ and $\mathcal{E}_{\{t,d\}}$ are the other types of across relations between joint of upper level information and both ego information, as well as underlying information, i.e., ''topic-words'' relations and ''topic-document'' relations.

We further illustrate the three types of relations (i.e., underlying relations $\mathcal{E}_{\{w,w\}}$, ego relations $\mathcal{E}_{\{w,d\}}$, upper level joint relations $\mathcal{E}_{\{t,w\}}$ and $\mathcal{E}_{\{t,d\}}$) over constructed text graph in details in the following.

1) UNDERLYING RELATIONS SET $\mathcal{E}_{\{w,w\}}$

Underlying relations over graph are the most essential relations exsiting in a document, for all the documents consist of a number of words. These words are with meaning by forming sentences, accordingly there are natural relations among different words. Such relations can help words with the similar meaning to pass information to each other through random walks during graph convolution process. This paper uses the point-wise mutual information (PMI) [30], [31] between two words to calculate the weight of the relations $e_{ij}^{\mathcal{W}} \in \mathcal{E}_{\{w,w\}}$. Specifically, we first set a text sliding window, and then use the sliding window to count the co-occurrence of words in the semantic space. Formally, we calculate the relations weights $e_{ij}^{\mathcal{W}}$ as follows:

$$e_{ij}^{\mathcal{W}} = \begin{cases} \mathrm{PMI}_{ij}, & \mathrm{PMI}_{ij} > \delta \\ 0, & \mathrm{PMI}_{ij} \leq \delta \end{cases}, \tag{1}$$



(a) Step I: Construction of MGTA Graphs

(b) Step II: Representation Learning with MGTA Graphs

FIGURE 3. The framework diagram of Text-MGNN. (a) An illustration of the text graph construction. (b) The illustration of prediction via GNNs.

where δ is a threshold of PMI used to filter out word edge weights whose mutual information is less than the threshold. PMI_{ij} is related to the co-occurrence frequency between w_i and w_j, the larger the co-occurrence frequency of these two words, the larger the value of PMI_{ij}. Formally, it is described as follows

$$\text{PMI}_{ij} = \log\left(\frac{p(w_i, w_j)}{(p(w_i)p(w_j))}\right), \tag{2}$$

where $p(w_i, w_j)$ is the probability that word w_i and word w_j appear in the sliding window at the same time. $p(w_i)$ and $p(w_j)$ are the probabilities of word w_i and word w_j appearing in the sliding window, respectively. Formally, it are described as follows

$$p(w_i, w_j) = \frac{Win(w_i, w_j)}{|Win|}, \tag{3}$$

where $Win(w_i, w_j)$ is the number of sliding windows that w_i and w_j appeared at the same time and $|Win|$ is the total number of sliding windows in the semantic spaces;

$$p(w_i) = \frac{Win(w_i)}{|Win|}, \tag{4}$$

where $Win(w_i)$ is the number of sliding windows that contain w_i.

2) EGO RELATIONS SET $\mathcal{E}_{\{w,d\}}$

Ego relations are at a higher level semantic representation than the underlying relations, the former can describe more deep relations to help document classification. From the perspective of the ego document, it is an abstract description of all words belonging to this document. Therefore, there is a natural connection between documents and words as well. But different words in a document have different contributions to the learning of its category-aware information. Some words without clear class attributes, such as prepositions and conjunctions, are relatively unimportant. This paper uses the Term Frequency-Inverse Document Frequency (TF-IDF) [32], [33] to represent the weight of the edge $e_{ij}^{\mathcal{W}\cup\Omega} \in \mathcal{E}_{\{w,d\}}$. Specifically, the frequency of the word existing in the document is counted at first, followed by counting the number of documents that include the word, and then the edge weight $e_{ij}^{\mathcal{W}\cup\Omega}$ is computed as the following.

$$e_{ij}^{\mathcal{W}\cup\Omega} = \begin{cases} TF_{(w_i,d_j)} \times IDF_{w_i}, & w_i \in d_j \\ 0, & w_i \notin d_j \end{cases}, \tag{5}$$

where $TF_{(w_i,d_j)}$ indicates the frequency of word w_i appearing in document d_j, and the larger the frequency, the higher the importance of the word w_i in the document d_i. In addition, the main idea of IDF_{w_i} is that if there are fewer documents containing the word w_i, the larger the IDF value, and then the better the category distinguishability of the word w_i.

Formally, it are described as follows

$$TF_{(w_i,d_j)} = \frac{N_{(w_i,d_i)}}{N_{|d_i|}}, \tag{6}$$

where $N_{(w_i,d_i)}$ is the frequency of word w_i in document d_j. $N_{|d_i|}$ is the total number of words in document d_j;

$$IDF_{w_i} = \log\left(\frac{N_\Omega}{(1+|\{j : w_i \in d_j\}|)}\right), \tag{7}$$

where $|\{j : w_i \in d_j\}|$ is the total number of document containing w_i.

3) UPPER LEVEL JOINT RELATIONS SET $\mathcal{E}_{\{t,w\}}$ AND $\mathcal{E}_{\{t,d\}}$

In order to obtain a more attribution-aware feature representation of each document, we further introduce topic information over graphs and build the relations set between topic and words, as well as topic and documents, referred respectively to as $\mathcal{E}_{\{t,w\}}$ and $\mathcal{E}_{\{t,d\}}$. Such relations designed on graphs can help enhance category distinguishability for features of documents; and then obtain good performance for document classification under category distinguishability representations. On the one hand, we propose a weighting scheme for describing the relations between words and topics, named class word frequency-inverse class frequency. One meaning of this type of relation can alleviate the influence for documents of those words without clear class information, such as conjunctions, prepositions, etc.; Another side can weaken the effect of words with multiple meanings on document class-aware information learning. Specifically, each edge $e_{ij}^{\mathcal{T}\bigcup\mathcal{W}} \in \mathcal{E}_{\{t,w\}}$ that is class word frequency-inverse class frequency between the topic node and word node can be described as

$$e_{ij}^{\mathcal{T}\bigcup\mathcal{W}} = \begin{cases} CTF_{(w_i,t_j)} \times ICF_{w_i}, & y_{w_i} = y_j \\ 0, & y_{w_i} \neq y_j \end{cases}, \tag{8}$$

where $CTF_{(w_i,t_j)}$ indicates the frequency of word w_i appearing in j-th topic t_j, and the larger the frequency, the higher the importance of the word w_i in the topic t_j. In addition, the main idea of ICF_{w_i} is that if there are fewer topics containing the word w_i, the larger the ICF value, and then the better the category distinguishability of the word w_i. Formally, it are described as follows

$$CTF_{(w_i,t_j)} = \frac{N_{(w_i,t_j)}}{N_{w\in t_j}}, \tag{9}$$

where $N_{(w_i,t_j)}$ is the frequency of word w_i in topic t_j and $N_{w\in t_j}$ is the number of words in j-th topic;

$$ICF_{w_i} = \log\left(\frac{C}{(1+|\{j : w_i \in t_j\}|)}\right), \tag{10}$$

where $|\{j : w_i \in t_j\}|$ is the total number of topic containing w_i.

On the other hand, the weighting scheme for the relations between documents and topics is a simple form, i.e., mean class word frequency of documents, which describes the information strengths of documents belonging to a certain category of topics. Specifically, relation $e_{ij}^{\mathcal{T}\bigcup\Omega}$ between each document node and topic node in $\mathcal{E}_{\{t,d\}}$ can be modeled as

$$e_{ij}^{\mathcal{T}\bigcup\Omega} = \begin{cases} MF_{(d_i,t_j)}, & y_{d_i} = y_j \\ 0, & y_{d_i} \neq y_j \end{cases}, \tag{11}$$

where $MF_{(d_i,t_j)}$ indicates the mean frequency of all words in document d_i appearing in the j-th topic. Formally, it is described as follows

$$MF_{(d_i,t_j)} = \frac{1}{N_{w\in d_i}} \sum_{w\in d_i} \left(CTF_{(w,t_j)} \times ICF_w\right), \tag{12}$$

where $N_{d\in t_j}$ is the number of documents in j-th topic. $N_{w\in d_i}$ denotes the number of words in document d_i.

B. FRAMEWORK OF TEXT-MGNN

For the given text data and its category label set, Text-MGNN first builds the text graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{X}\}$ by the proposed method as mentioned in Section IV-A. And obtain adjacency matrix $\mathbf{A}$ by $\mathcal{E}$ for further graph learning, which is shown in Figure 3 (b). Specifically, $\mathbf{A} \in \mathbb{R}^{N\times N}$, where $N = N_\Omega + N_\mathcal{W} + C$. Each element a_{ij} of $\mathbf{A}$ is equal to e_{ij}, where $e_{ij} \in e^\mathcal{W} \bigcup e^{\mathcal{W}\bigcup\Omega} \bigcup e^{\mathcal{T}\bigcup\mathcal{W}} \bigcup e^{\mathcal{T}\bigcup\Omega}$ and $e^\mathcal{S}$ is the set of all relations $e_{ij}^\mathcal{S}$, $\mathcal{S} \in \{\mathcal{W}, \mathcal{W}\bigcup\Omega, \mathcal{T}\bigcup\mathcal{W}, \mathcal{T}\bigcup\Omega\}$. Accordingly, the text graph is further defined as $\mathcal{G} = \{\mathcal{V}, \mathbf{A}, \mathbf{X}\}$. Then, use one kind of GNNs (e.g., GCN [34], GAT [35]) to implement text classification. Formally, we define the l-th layer of Text-MGNN as

$$\mathbf{m}_v^{(l)} = \mathsf{AGGREGATE}_{\theta_{l_1}}^{(l)}\left(\{\mathbf{h}_u^{(l-1)} : u \in \mathcal{N}(v)\}\right), \tag{13}$$

$$\mathbf{h}_v^{(l)} = \mathsf{UPDATE}_{\theta_{l_2}}^{(l)}\left(\mathbf{h}_v^{(l-1)}, \mathbf{m}_v^{(l)}\right), \tag{14}$$

where $\mathsf{AGGREGATE}_{\theta_{l_1}}^{(l)}(\cdot)$ and $\mathsf{UPDATE}_{\theta_{l_2}}^{(l)}(\cdot)$ are the aggregate functions and update functions of l-th layer graph convolution operation with parameters θ_{l_1} and parameters θ_{l_2}. The whole parameters of Text-MGNN can be defined as $\theta = \{\theta_{l_1}, \theta_{l_2}\}_{l=1}^L$ if we conduct L times graph convolution operation.

C. OPTIMIZATION OBJECTIVE

After obtaining more distinguishable representations of document nodes, we validate the performance of Text-MGNN on text classification task, accordingly, a cross-entropy loss is used for training [36]. The objective function of whole framework can be described as

$$\mathcal{L}_\theta = -\sum_{d\in\Omega_{\text{train}}} y_d \ln z_d, \tag{15}$$

where Ω_{train} is the set of training document nodes, y_d is the label indicator vector of document node d. The whole conducting process of text classification task by Text-MGNN is summarised in Algorithm 1.

Remark: The upper level joint relations are the core component of this paper, which can effectively alleviate the propagation of heterophily information among documents caused by polysemy words. Specifically, it not only builds

Algorithm 1 Text-MGNN Algorithm

Require: Training data;
Ensure: Return best parameters θ;
1: **Initialize:** Minimum term frequency, sliding window size, and initial parameters θ;
2: For texts (i.e., documents and words) in the training set, conduct a series of preprocessing operations at first, such as, word segmentation, case changer, stop word removal, etc., to obtain the initial feature matrix $\mathbf{X}$;
3: Constructing the topic-aware text graph $\mathcal{G} = \{\mathcal{V}, \mathbf{A}, \mathbf{X}\}$ by modeling the multi-granular relations $e_{ij}^{\mathcal{W}}$, $e_{ij}^{\mathcal{W}\bigcup\Omega}$, $e_{ij}^{\mathcal{T}\bigcup\mathcal{W}}$ and $e_{ij}^{\mathcal{T}\bigcup\Omega}$ of $\mathcal{G}$ following (1), (5), (8) and (11) respectively;
4: **while** not done **do**
5: Learn category representations $\mathbf{z}$ of $\mathbf{X}$ over text graphs $\mathcal{G}$ using one kind of GNNs model with initial parameters θ as (13) and (14);
6: Calculate loss $\mathcal{L}_\theta$ by (15);
7: Update the parameters θ with

$$\theta = \arg\min_{\theta} \mathcal{L}_\theta$$

8: **end while**

the relations between documents and topics, but also words and topics. The topics information can be enchanced for these documents by the relation between documents and topics on the one hand. On the other hand, the enhancing of words and topic information can also further strengthen the relations between documents and topics though the relations between words and documents.

V. EXPERIMENTS

In this section, we conduct extensive experiments to evaluate the performance of Text-MGNN. Specifically, we try to answer the following questions:

Q1: How does the proposed Text-MGNN perform on text classification compared with three types of mainstream methods? (Section V-E)

Q2: Does the introduction of topics enhance text classification? (Section V-F)

Q3: How does the key parameters in constructed text graph influence the performance of Text-MGNN? (Section V-H)

A. DATASETS

To comprehensively evaluate the performance of Text-MGNN, we conduct experiments on four benchmark datasets, which are R8 dataset, MR dataset, R52 dataset and Ohsumed dataset. R8 dataset and R52 dataset is two subsets of Reuters 21, 578 datasets, which is applied to multi-class task. R8 dataset is classified in to 8 categories, totally contains with 5, 485 documents for training and 2, 189 documents for testing. R52 dataset has total 52 categories and is divided into 6, 532 training samples and 2, 568 testing samples. MR dataset is a short text dataset, which is a movie review dataset for binary sentiment classification. The corpus has 5, 331 positive reviews and 5, 331 negative reviews. The ohsumed dataset contains 7, 400 documents categorized into 23 different categories and was split into 3, 357 training and 4, 043 test documents. The summary of the datasets is given in Table 1.

B. BASELINES

We aim to evaluate the performance of Text-MGNN on text classification task against three types of text classification models. One is text classification methods by classical neural network, such as TF-IDF + LR [37], CNN [9], LSTM [11]. Second is compared with methods based on word embedding, i.e., PV-DBOW [38], PV-DM [38], fastText [39], SWEM [40] and LEAM [41]. The final kind of baseline is based on graph learning, e.g., TextGCN [15], SGC [27] and Graph-CNN [42].

C. EVALUATION METRICS

- The accuracy (Accuracy) is used as an index to evaluate the performance of text classification in this paper and can be described as

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \tag{16}$$

  where TP, TN, FP and FN are the number of true positives, true negatives, false positives and false negatives, respectively.
- The precision (Precision) stands for the ratio of the number of correctly predicted positives to the predicted positives. It can be computed as

$$\text{Precision} = \frac{TP}{TP + FP}. \tag{17}$$

- The recall (Recall) calculates the proportion of positives (TP) that are correctly predicted to all positives in the true label sample. This metric can be written as

$$\text{Recall} = \frac{TP}{TP + FN}. \tag{18}$$

- The f-measure (F-measure) is calculated by Precision and Recall, which can be written as

$$\text{F-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{19}$$

D. EXPERIMENT SETUP

For experimental models, the initial features for words and documents are denoted as a identity matrix, which is a common operation accorrding to the classical methods, such as [15]. In addition, accorrding to the parameter analysis, we define the model with two layers, and use Adam optimizer [43] for model optimization, where learning rate is set to 0.02. Other hype-parameters in model, such as dropout and hidden dimension is set to 0.5 and 200, respectively. The training stop condition is that the current batch validation set loss is greater than the average loss of

TABLE 1. Statistics of the Datasets.

Dataset	Docs	Training	Test	Words	Nodes	Classes	Average Length
R8	7,674	5,485	2,189	7,688	15,362	8	65.72
MR	10,662	7,108	3,554	18,764	29,426	2	20.39
R52	9,100	6,532	2,568	8,892	18,044	52	69.82
Ohsumed	7,400	3,357	4,043	14,157	21,557	23	135.82

TABLE 2. Performance on Text-MGNN against the methods based on word embedding. The highlighted as bold is denoted the best model, and the second best is underlined.

Method	R8	R52	MR	Ohsumed
PV-DBOW	0.8587	0.7829	0.6109	0.4665
PV-DM	0.5207	0.4492	0.5947	0.2950
fastText	0.9613	0.9281	0.7514	0.5770
SWEM	0.9532	0.9294	0.7665	0.6312
LEAM	0.9331	0.9184	0.7695	0.5858
Text-MGNN	**0.9739**	**0.9420**	**0.7746**	**0.7000**
Improving	1.26%	1.26%	0.51%	6.88 %

TABLE 3. Performance on Text-MGNN against text classification methods by classical neural network. The highlighted as bold is denoted the best model, and the second best is underlined.

Method	R8	R52	MR	Ohsumed
TF-IDF + LR	0.9374	0.8695	0.7459	0.5466
CNN	0.9402	0.8537	0.7498	0.4387
LSTM	0.9368	0.8554	0.7506	0.4113
Text-MGNN	**0.9739**	**0.9420**	**0.7746**	**0.7000**
Improving	3.37%	7.25%	2.40%	15.34%

TABLE 4. Performance on Text-MGNN against baseline is based on graph learning. The highlighted as bold is denoted the best model, and the second best is underlined.

Dataset	Graph-CNN	TextGCN	SGC	Text-MGNN	Improving
R8	0.9689	0.9707	0.9720	**0.9739**	0.19%
R52	0.9320	0.9356	0.9400	**0.9420**	0.20%
MR	0.7674	0.7674	0.7590	**0.7746**	0.72%
Ohsumed	0.6386	0.6836	0.6850	**0.7000**	1.5%

the previous 10 batches. For the baseline model, we adopt the best parameters provided in the original paper. In addition, in order to ensure the fairness of the ablation study, under the parameters setting of the model training phase as above mentioned, we further set the parameters that may affect the experiment results uniformly. Firstly, during the word removal stage, we set the minimum word frequency to 5, that is, words with a word frequency less than 5 in the dataset are not used as nodes on the text graph. Secondly, during the composition stage, the PMI threshold is set to 0, i.e., if the edge weight between two nodes is less than 0, it is considered that there is no connection between the two nodes, and finally, the size of the word co-occurrence sliding window is set to 20.

E. PERFORMANCE ON TEXT CLASSIFICATION

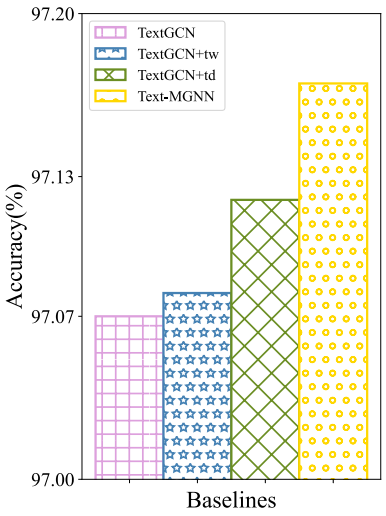
The performance results between Text-MGNN and diverse baseline methods are compared in Table 2 - Table 4. From the results in all table, Text-MGNN achieves the top accuracy compared to all baselines. In addition, it is worth noting that Text-MGNN achieve the greatest improvement at least 7.05 percentage point on R52 dataset compared with the methods based on classical neural network for text classification. One reasonable explansion is that these classical neural networks for text classification are difficult to learn the latent associations among documents, while the baselines based on graph learning can learn the interactive features among documents by building the relation graphs between documents and words, as well as among words. And from the result, the baselines based on graph learning significantly outperform the other two types of baselines. This result shows that methods based on graph learning will bring great potential to text classification.

F. EFFECT OF UPPER LEVEL JOINT RELATIONS

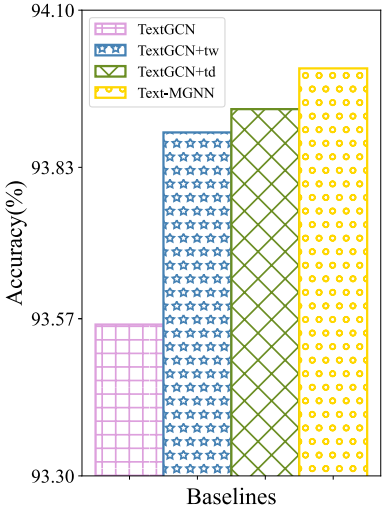
Upper-level joint relations are the core components of Text-MGNN, which is a combination relation containing relations between topic and words, as well as topic and documents. To verify the promoting effect of two kinds of relations on text classification, we design three sets of comparison ablation experiments, which include four baselines used to validate the result. ‘‘TextGCN’’ is the basic baseline used to validate the devised components of Text-MGNN. The second baseline is denoted as ‘‘TextGCN+td’’, which means adding the relation between the topic and documents based on TextGCN. The third baseline ‘‘TextGCN+tw’’ adds the relation between the topic and words based on TextGCN. The last baseline is called ‘‘Text-MGNN’’ proposed in this paper, which means adding the relations at the same time between topic and documents, as well as topic and words. These baselines are compared on all the datasets. The results are plotted in Figure 4.

1) RELATIONS BETWEEN THE TOPIC AND TEXT CAN ENHANCE TEXT CLASSIFICATION

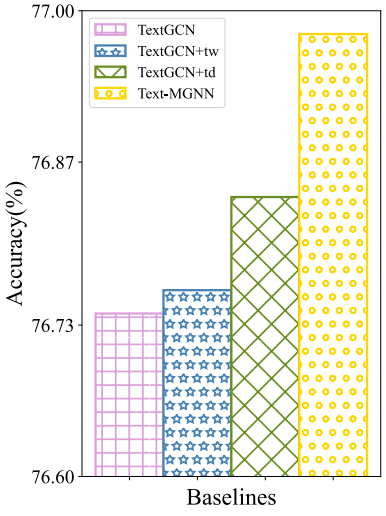
Comparing the performance of ‘‘TextGCN’’ and ‘‘TextGCN +td’’ in Figure 4, we can draw a conclusion that relations between the topic and documents can help promote the performance of text classification. On the other hand, relations between the topic and words can also help promote the performance of text classification can be observed by comparing the performance of ‘‘TextGCN’’ and ‘‘TextGCN+tw’’. One reasonable explanation for this is the introduction of topic information can facilitate class-related information learning of the text nodes (i.e., document nodes and word nodes), which is conducive to improving the performance of text classification.



(a) R8



(b) R52



(c) MR

FIGURE 4. Effects of upper-level joint relations for text classification.

TABLE 5. The performance improvement on TextGCN and SGC with the help of the MGTA graph. The highlighted as **bold** is denoted the best model.

Dataset	Metric	TextGCN	TextGCN w. MGTA	SGC	SGC w. MGTA
R8	Precision	0.9439	**0.9575**	0.9358	**0.9577**
	Recall	0.9232	**0.9317**	0.9177	**0.9396**
	F-measure	0.9315	**0.9412**	0.9267	**0.9486**
	Accuracy	0.9707	**0.9739**	0.9720	**0.9767**
R52	Precision	0.7648	**0.8016**	0.7469	**0.7600**
	Recall	0.6870	**0.6909**	0.6872	**0.7027**
	F-measure	0.7082	**0.7197**	**0.7158**	0.7027
	Accuracy	0.9356	**0.9420**	0.9400	**0.9408**
MR	Precision	0.7553	**0.7774**	0.7629	**0.7746**
	Recall	0.7538	**0.7746**	0.7622	**0.7743**
	F-measure	0.7534	**0.7741**	0.7626	**0.7745**
	Accuracy	0.7674	**0.7746**	0.7590	**0.7743**
Ohsumed	Precision	**0.7152**	0.7026	**0.6606**	0.6577
	Recall	0.5983	**0.6103**	0.5966	**0.6138**
	F-measure	0.6298	**0.6377**	0.6269	**0.6350**
	Accuracy	0.6836	**0.7000**	0.6850	**0.6968**

* The accuracy of TextGCN and SGC comes from the original papers [15] and [27], other results are reproduced according to the experimental environment of this paper.

2) TEXT-MGNN HAS A BETTER PERFORMANCE VIA BUILDING UPPER LEVEL JOINT RELATIONS

Another important result should be noted that "Text-MGNN" can further achieve breakthrough performance compared with "TextGCN+td" and "TextGCN+tw" by combining the above two relations. In addition, it also can be seen that the performance of "Text-MGNN" is greater than that of "TextGCN" on the three datasets.

G. FLEXIBILITY OF TEXT-MGNN

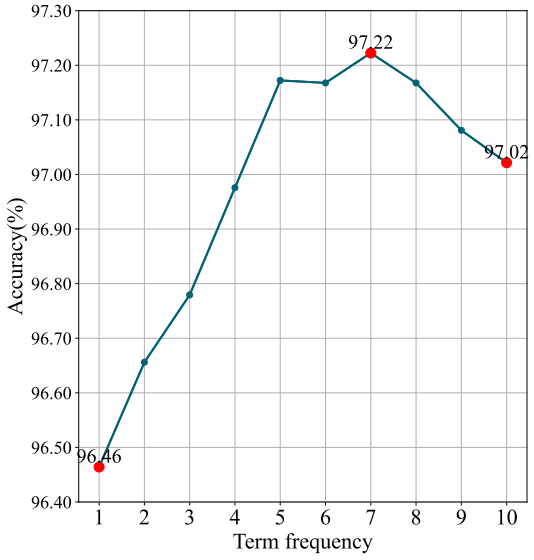
Text-MGNN is flexible. Its core step is the construction of MGTA graphs, and such MGTA graph is applicable to other graph neural networks-based text classification models, such as SGC and TextGCN. Table 5 shows the comparison results on four metrics between TextGCN with MGTA graphs referred to as TextGCN w. MGTA and TextGCN, as well as SGC with MGTA graphs abbreviated as SGC w. MGTA and SGC. In general, we focus on accuracy, which is the common metric of the classification task. From the results, it can be observed that the model with the help of MGTA graphs can effectively improve the performance.

H. PARAMETERS STUDY

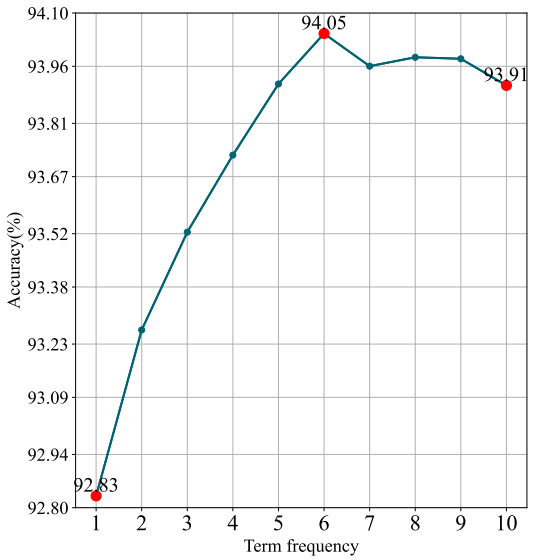
Two main parameters in Text-MGNN are minimum term frequency, and sliding window size. The minimum term frequency, as a threshold for filtering keywords, can determine the size of the graph. The sliding window size is used to achieve the statistics of word co-occurrence, which is a key value of preprocessing way for modeling the relations among words. The joint effect of these two parameters can basically determine the structure of the graph. In the following, we use the control variates to explore the most suitable parameters.

1) PARAMETERS STUDY OF MINIMUM TERM FREQUENCY

The minimum term frequency range is set as an integer from 1 to 10 in this paper following other classical

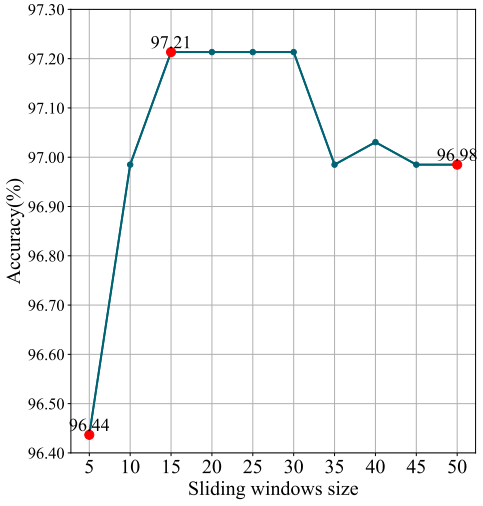


(a) R8

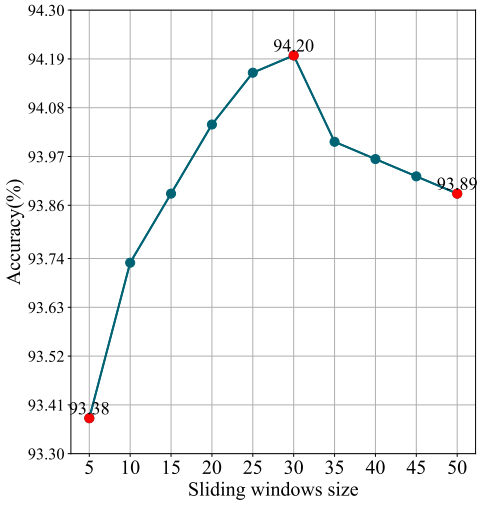


(b) R52

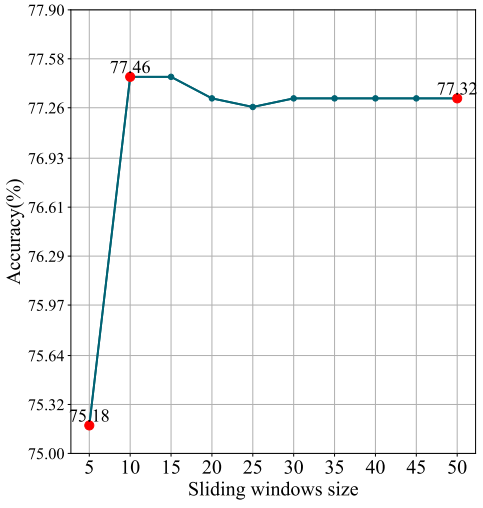
FIGURE 5. The effect of minimum term frequency for text classification.



(a) R8



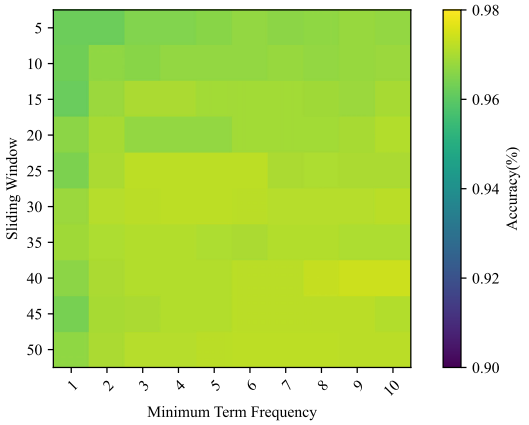
(b) R52



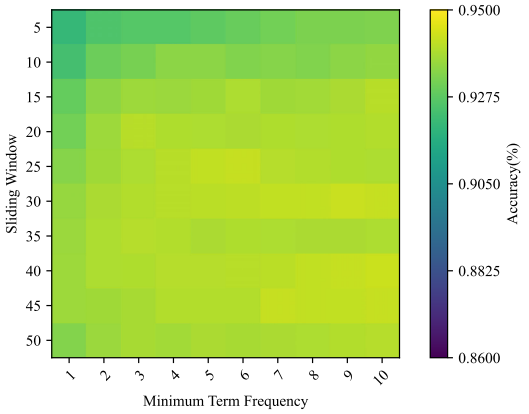
(c) MR

FIGURE 6. The effect of sliding window size for text classification.

methods [44]. In general, the greater the minimum term frequency, the stronger the connection between the word nodes and topics in the graph. However, too large a term frequency will also ignore the detailed description of documents, making it difficult to distinguish documents of similar topics. This problem commonly occurs in short document data. For this reason, minimum term frequency is usually set to 1 in short text data, e.g., MR dataset. As illustrated in Figure 5, the results in Figure 5 (a) and Figure 5 (b) demonstrate a result with a common phenomenon on long document data that the classification performance will be improved with the increase of the minimum term frequency, this may be because some words that are lower relevant to class-aware representation learning are filtered out. And then, there is a certain degree of performance decline after reaching a certain level. This may be because there are too many filtered words and then some

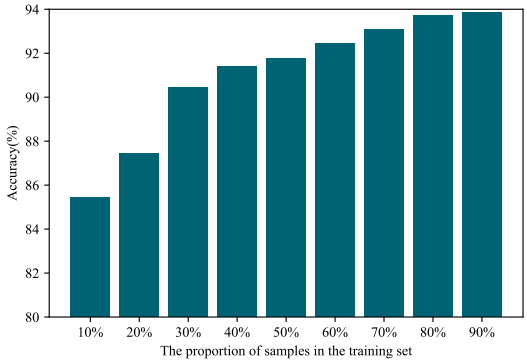


(a) R8

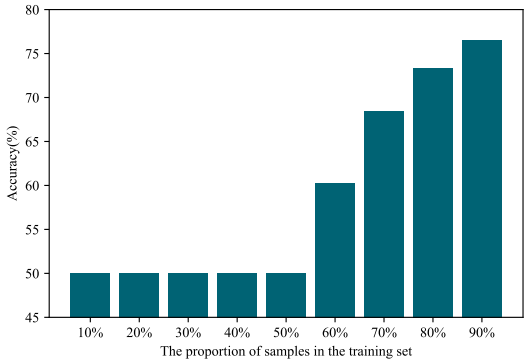


(b) R52

FIGURE 7. The joint effect of minimum term frequency and sliding window size for text classification.



(a) R52



(b) MR

FIGURE 8. Performance of different proportions of samples in train dataset of R52 dataset and MR dataset.

documents with few words may become isolated nodes, while the keywords reserved in the text graph are difficult to learn the more detailed description of documents. In addition, it can also be observed that when the minimum term frequency is 1, which is a particular situation that all valid words of documents remain in text graphs, the performance is poor on all datasets. This means that there may be words lower relevant to the class-aware representation learning in the text graph.

2) PARAMETERS STUDY OF SLIDING WINDOW SIZE

The sliding window is used to control the number of relations among words. The larger size of the sliding window, the more words are within the same sliding window, and the more contextual semantic relations are modeled. But the too-large size of the sliding window may make the semantic relations between irrelevant phrases be modeled, thus introducing too much noisy information. We, referred to [45], take an integer range from 5 and 50 with a step 5 in turn to discuss the influence of sliding window size. The result is shown in Figure 6, from which can be observed that the performance will gradually improve with the increase of the sliding window size, and then there will be a certain degree of decline or basically stabilization.

3) TWO PARAMETER JOINT ANALYSIS

The combined effect of the above-mentioned parameters will have a decisive impact on the text graph construction. In order to further observe the effect of these two parameters on classification performance simultaneously, we conduct more extensive parameter experiments. The results are shown in Figure 7, which illustrates a parameter influence heat map of text classification performance on R8 dataset and R52 dataset. Each color block represents the classification accuracy obtained by the parameter value corresponding to its horizontal and vertical coordinates. The brighter the color, the higher the accuracy. From the result of Figure, we can seen that on the one hand, the impact of these two parameters on classification performance is stable, and on the other hand, the larger the minimum term frequency and the sliding window size are, the better the performance will be.

4) DIFFERENT PROPORTIONS OF TRAINING SAMPLES ANALYSIS

To demonstrate various validation results, we have conducted a series of comparison experiments on nine different

TABLE 6. Training and inference times of TextGCN and Text-MGNN on four datasets.

Dataset	TextGCN			Text-MGNN		
	Training Time(s)	Inference Time(s)	Accuracy	Training Time(s)	Inference Time(s)	Accuracy
R8	0.2078	0.0598	0.9707	0.2334	0.0647	0.9739
R52	0.3004	0.0862	0.9356	0.3446	0.1001	0.9420
MR	0.9949	0.3626	0.7674	1.0145	0.3914	0.7746
Ohsumed	0.4027	0.1181	0.6836	0.4403	0.1258	0.7000

TABLE 7. Some notations used in the paper.

Notations	Description
Ω	Document corpus
$\mathcal{W}$	The set of words
$\mathcal{T}$	The set of topics
$\mathcal{Y}$	The set of label
$w_i, i = 1, 2, 3, \ldots, N_{\mathcal{W}}$	The i-th word
$d_i, i = 1, 2, \ldots, N_{\Omega}$	The i-th document
$y_i, i = 1, 2, 3, \ldots, C$	The i-th label
$t_i, i = 1, 2, 3, \ldots, C$	The one-hot representation for the label of i-th document
y_{d_i}	The label of document d_i
$w_{i_j}, j = 1, 2, \ldots, i_{\lvert d_i \rvert}$	The j-th word of document d_i
N_{Ω}	The number of documents
$N_{\mathcal{W}}$	The number of words
C	The total number of categories
$\mathcal{G}$	Text graph
$\mathcal{V}$	The node set of text graph
$\mathcal{E}$	The edge set of text graph
N_n	The total number of node types
N_e	The total number of edge types
$\mathcal{V}_i$	The i-th type of node set
$\mathcal{E}_i$	The i-th type of edge set
$\mathbf{X}$	Node feature matrix
$\mathcal{E}_{\{w,w\}}$	The relation set among words
$\mathcal{E}_{\{w,d\}}$	The relation set between words and documents
$\mathcal{E}_{\{t,w\}}$	The relation set between topics and words
$\mathcal{E}_{\{t,d\}}$	The relation set between topics and documents
$e_{ij}^{\mathcal{W}}$	The relation between w_i and w_j
$e_{ij}^{\mathcal{W} \bigcup \Omega}$	The relation between w_i and d_j
$e_{ij}^{\mathcal{T} \bigcup \mathcal{W}}$	The relation between w_i and t_j
$e_{ij}^{\mathcal{T} \bigcup \Omega}$	The relation between d_i and t_j
PMI_{ij}	Point mutual information value between w_i and w_j
$Win(w_i)$	The number of times w_i occurs in the sliding window
$\lvert Win \rvert$	The total number of sliding window
$p(w)$	The probability of w appearing in the sliding window
$p(w_i, w_j)$	The probability of w_i and w_j appearing in the sliding window at the same time
$TF_{(w_i,d_j)}$	The probability of w_i appearing in the d_j
IDF_{w_i}	The inverse document frequency of w_i
$N_{(w_i,d_i)}$	The number of times w_i occurs in d_j
$N_{\lvert d_i \rvert}$	The total number of words in i-th document
$CTF_{(w_i,t_j)}$	The frequency of w_i appearing in the j-th topics
ICF_{w_i}	The inverse class frequency of w_i
$N_{(w_i,t_j)}$	The number of occurrences of w_i in the j-th type of document
$N_{w \in t_j}$	The total number of words belonging to j-th topics
$\mathbf{A}$	The adjacency matrix

proportions of training samples from train datasets, that is, ranging from 10% to 90% with the step of 10%. It should be noted that if selecting 90% samples as train samples, then other 10% samples in train sets are validation samples. All results are shown in Figure 8. It can be found that the performance of the model improves with the increase in the number of training samples, which is natural for more training samples can provide richer category information. Among them, the way of selecting 90% training samples is the common validation method for most text classification models, so we continue to use this validation method for comparison with other methods in this paper.

I. TIME CONSUMPTION

We provide the time consumption statistics of Text-MGNN, which is listed in Table 6. Compared with TextGCN, the triple nodes will cause an increase in the time consumption of Text-MGNN. In theory, compared with TextGCN, the training time of Text-MGNN will increase with the increase of topic nodes. This is because triple nodes will provide a larger size of text

tensor in the matrix operation of graph convolution, which will require more parameters for graph learning. However, the number of topic nodes is at most the number of categories in the dataset, which limits the infinite expansion of the text tensor size. It can be observed from Table 6 that the R52 dataset, which is the dataset with the most categories used in this paper, only increases the time consumption of a batch by 0.0442s in training, but achieves performance improved. In addition, there is little difference in the inference time between the two models on four datasets.

VI. CONCLUSION

This paper presented a GNN method for text classification from multi-granular topic-aware perspective, referred to as Text-MGNN. Text-MGNN can enhance text classification by alleviating the effect of heterophily information under the representation learning via GNNs caused by polysemy words. The results from various experiments show the excellent performance of Text-MGNN over several real-world text classification datasets, as well as validate the components effectiveness and discuss key paramsters of Text-MGNN. Future work can focus on the representation learning of dynamic text graphs under the problem of polysemy words.

APPENDIX. NOTATIONS

Here we list the important symbols in this paper, which are summarized in Table 7.

REFERENCES

[1] M. Birjali, M. Kasri, and A. Beni-Hssane, "A comprehensive survey on sentiment analysis: Approaches, challenges and trends," *Knowl.-Based Syst.*, vol. 226, Aug. 2021, Art. no. 107134.

[2] A. Gandhi, K. Adhvaryu, S. Poria, E. Cambria, and A. Hussain, "Multimodal sentiment analysis: A systematic review of history, datasets, multimodal fusion methods, applications, challenges and future directions," *Inf. Fusion*, vol. 91, pp. 424–444, Mar. 2023.

[3] P. Kaur, H. S. Pannu, and A. K. Malhi, "Comparative analysis on cross-modal information retrieval: A review," *Comput. Sci. Rev.*, vol. 39, Feb. 2021, Art. no. 100336.

[4] I. Amin and M. K. Dubey, "An overview of soft computing techniques on review spam detection," in *Proc. 2nd Int. Conf. Intell. Eng. Manage. (ICIEM)*, Apr. 2021, pp. 91–96.

[5] K. Kowsari, K. J. Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, "Text classification algorithms: A survey," *Information*, vol. 10, no. 4, p. 150, Apr. 2019.

[6] M. Eminagaoglu, "A new similarity measure for vector space models in text classification and information retrieval," *J. Inf. Sci.*, vol. 48, no. 4, pp. 463–476, Aug. 2022.

[7] I. Lauriola, A. Lavelli, and F. Aiolli, "An introduction to deep learning in natural language processing: Models, techniques, and tools," *Neurocomputing*, vol. 470, pp. 443–456, Jan. 2022.

[8] A. Jacovi, O. S. Shalom, and Y. Goldberg, "Understanding convolutional neural networks for text classification," in *Proc. EMNLP Workshop BlackboxNLP, Analyzing Interpreting Neural Netw. NLP*, 2018, pp. 56–65.

[9] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. Conf. Empirical Methods Natural Lang. Process.* Doha, Qatar: Association for Computational Linguistics, Apr. 2014, pp. 1746–1751.

[10] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 1–7.

[11] P. Liu, X. Qiu, and X. Huang, "Recurrent neural network for text classification with multi-task learning," in *Proc. 25th Int. Joint Conf. Artif. Intell.*, 2016, pp. 2873–2879.

[12] Z. Xie, W. Lv, S. Huang, Z. Lu, B. Du, and R. Huang, "Sequential graph neural network for urban road traffic speed prediction," *IEEE Access*, vol. 8, pp. 63349–63358, 2019.

[13] Z. Ye, Y. J. Kumar, G. O. Sing, F. Song, and J. Wang, "A comprehensive survey of graph neural networks for knowledge graphs," *IEEE Access*, vol. 10, pp. 75729–75741, 2022.

[14] Z. Xing and S. Tu, "A graph neural network assisted Monte Carlo tree search approach to traveling salesman problem," *IEEE Access*, vol. 8, pp. 108418–108428, 2020.

[15] L. Yao, C. Mao, and Y. Luo, "Graph convolutional networks for text classification," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 7370–7377.

[16] S. S. Birunda and R. K. Devi, "A review on word embedding techniques for text classification," *Innovative Data Communication Technologies and Application*. 2021, pp. 267–281.

[17] Y. Liu, P. Li, and X. Hu, "Combining context-relevant features with multi-stage attention network for short text classification," *Comput. Speech Lang.*, vol. 71, Jan. 2022, Art. no. 101268.

[18] X. Liu, X. You, X. Zhang, J. Wu, and P. Lv, "Tensor graph convolutional networks for text classification," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, 2020, pp. 8409–8416.

[19] A. W. Haryanto, E. K. Mawardi, and Muljono, "Influence of word normalization and chi-squared feature selection on support vector machine (SVM) text classification," in *Proc. Int. Seminar Appl. Technol. Inf. Commun.*, Sep. 2018, pp. 229–233.

[20] H. Gao, X. Zeng, and C. Yao, "Application of improved distributed naive Bayesian algorithms in text classification," *J. Supercomput.*, vol. 75, no. 9, pp. 5831–5847, Sep. 2019.

[21] K. Shah, H. Patel, D. Sanghvi, and M. Shah, "A comparative analysis of logistic regression, random forest and KNN models for the text classification," *Augmented Hum. Res.*, vol. 5, no. 1, pp. 1–16, 2020.

[22] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *Proc. Eur. Conf. Mach. Learn.* Cham, Switzerland: Springer, 1998, pp. 137–142.

[23] S. I. Wang and C. D. Manning, "Baselines and bigrams: Simple, good sentiment and topic classification," in *Proc. 50th Annu. Meeting Assoc. Comput. Linguistics*, 2012, pp. 90–94.

[24] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Proc. 28th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, vol. 1. Cambridge, MA, USA: MIT Press, 2015, pp. 649–657.

[25] K. Sinha, Y. Dong, J. C. K. Cheung, and D. Ruths, "A hierarchical neural attention-based text classifier," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 817–823.

[26] B. Xu, K. Cen, J. Huang, H. Shen, and X. Cheng, "A survey on graph convolutional neural network," *Chin. J. Comput.*, vol. 43, no. 5, pp. 755–780, 2020.

[27] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger, "Simplifying graph convolutional networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6861–6871.

[28] A. Li, Z. Qin, R. Liu, Y. Yang, and D. Li, "Spam review detection with graph convolutional networks," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, Nov. 2019, pp. 2703–2711.

[29] D.-H. Pham and A.-C. Le, "Exploiting multiple word embeddings and one-hot character vectors for aspect-based sentiment analysis," *Int. J. Approx. Reasoning*, vol. 103, pp. 1–10, Dec. 2018.

[30] K. W. Church and P. Hanks, "Word association norms, mutual information, and lexicography," *Comput. Linguistics*, vol. 16, no. 1, pp. 22–29, Mar. 1990.

[31] P. R. Kanna and P. Pandiaraja, "An efficient sentiment analysis approach for product review using Turney algorithm," *Proc. Comput. Sci.*, vol. 165, pp. 356–362, Jan. 2019.

[32] H. C. Wu, R. W. P. Luk, K. F. Wong, and K. L. Kwok, "Interpreting TF-IDF term weights as making relevance decisions," *ACM Trans. Inf. Syst.*, vol. 26, no. 3, pp. 1–37, Jun. 2008.

[33] A. Dhar, H. Mukherjee, N. S. Dash, and K. Roy, "Text categorization: Past and present," *Artif. Intell. Rev.*, vol. 54, pp. 3007–3054, Sep. 2021.

[34] M. Welling and T. N. Kipf, "Semi-supervised classification with graph convolutional networks," in *Proc. ICLR*, 2016, pp. 1–14.

[35] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proc. ICLR*, 2018, pp. 1–12.

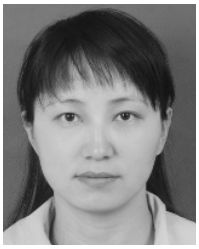
[36] P. Golik, P. Doetsch, and H. Ney, "Cross-entropy vs. squared error training: A theoretical and experimental comparison," in *Proc. Interspeech*, vol. 13, Aug. 2013, pp. 1756–1760.

[37] Z. Rezaei, B. Eslami, M.-A. Amini, and M. Eslami, "Hierarchical three-module method of text classification in web big data," in *Proc. 6th Int. Conf. Web Res. (ICWR)*, Apr. 2020, pp. 58–65.
[38] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1188–1196.
[39] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics*, vol. 2. Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 427–431.
[40] D. Shen, G. Wang, W. Wang, M. R. Min, Q. Su, Y. Zhang, C. Li, R. Henao, and L. Carin, "Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms," in *Proc. 56th Ann. Meeting Assoc. Comput. Linguistics*. Melbourne, VIC, Australia: Association for Computational Linguistics, Jul. 2018, pp. 440–450.
[41] G. Wang, C. Li, W. Wang, Y. Zhang, D. Shen, X. Zhang, R. Henao, and L. Carin, "Joint embedding of words and labels for text classification," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*. Melbourne, VIC, Australia: Association for Computational Linguistics, 2018, pp. 2321–2331.
[42] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 3844–3852.
[43] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. Process. Int. Conf. Learn. Represent.*, 2015, pp. 1–15.
[44] A. I. Kadhim, "Survey on supervised machine learning techniques for automatic text classification," *Artif. Intell. Rev.*, vol. 52, no. 1, pp. 273–292, Jun. 2019.
[45] L. Han, T. Finin, P. McNamee, A. Joshi, and Y. Yesha, "Improving word similarity by augmenting PMI with estimates of word polysemy," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 6, pp. 1307–1322, Jun. 2013.

YONGCHUN GU received the M.Sc. degree in applied mathematics from China Jiliang University, Hangzhou, China, in 2021. He is currently a teaching Assistant with the Sichuan University of Arts and Sciences, Dazhou, China. His research interests include machine learning, natural language processing, and graph neural networks.

YI WANG received the M.Sc. degree in applied mathematics from China Jiliang University, Hangzhou, China, in 2021. She is currently pursuing the Ph.D. degree in computer science with Zhejiang Normal University, Jinhua, Zhejiang, China. Her research interests include deep learning and graph processing.

HENG-RU ZHANG received the M.S. degree from the School of Mechanical and Electrical Engineering, University of Electronic Science and Technology of China, Chengdu, China, in 2002, and the Ph.D. degree from the School of Sciences, Southwest Petroleum University, Chengdu, in 2019. He is currently a Professor with Southwest Petroleum University. He has published more than 30 refereed papers in various journals and conferences, including *Pattern Recognition*, *Information Sciences*, and *Knowledge-Based Systems*. His current research interests include recommender systems, label distribution learning, and granular computing.



JIAO WU (Member, IEEE) received the Ph.D. degree from the School of Computer Science, Xidian University, Xi'an, in 2012. She is currently an Associate Professor with China Jiliang University. She has published some articles in journals and international conferences, including IEEE TRANSACTIONS ON IMAGE PROCESSING. Her current research interests include text data mining, machine learning, and compressed sensing.

XINGQUAN GU received the M.S. degree from Zhejiang University, Hangzhou, China, in 2005. He is currently an Associate Professor with China Jiliang University. He has published several articles in various journals and conferences. His current research interest includes standardized text data mining.