Soft Computing (2024) 28:8119–8132
https://doi.org/10.1007/s00500-024-09679-y

APPLICATION OF SOFT COMPUTING

Topic-aware cosine graph convolutional neural network for short text classification

Changrong Min[1] · Yonghe Chu[3] · Hongfei Lin[2] · Bolin Wang[2] · Liang Yang[2] · Bo Xu[2]

Accepted: 15 January 2024 / Published online: 3 July 2024
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2024

Abstract

Graph Convolutional Network (GCN) has been extensively studied in the task of short text classification (STC), utilizing global graphs that incorporate texts at different levels of granularity to learn text embeddings. However, the GCN-based methods only focus on the alignment between ground-truth labels and predicted labels, overlooking the geometric structure implicitly encoded by the graph. To address this limitation, we propose a novel GCN-based method that is entitled Topic-aware Cosine GCN (ToCo-GCN) for the STC. The ToCo-GCN defines and captures underlying geometric structures of short texts from different categories in the cosine space. Specifically, the ToCo-GCN regards the within-class and between-class geometric structures as constraint, aiming to learn both representative and discriminative short text representations. Moreover, to mitigate the inherent sparsity problem of short texts, the ToCo-GCN augment the text graph with latent topics. Experimental results on 8 STC datasets demonstrate that the ToCo-GCN is superior to state-of-the-art baselines in terms of Accuracy and Macro-F1 score.

Keywords Graph convolutional network · Short text classification · Discriminative learning · Topic models

1 Introduction

With the rapid development of e-commerce and social media platforms, users are generating a large volume of short texts on a daily basis, including product reviews and online forum posts, among others.This significant increase in short texts on the web has led to a growing interest in the STC task from both industry and academia. The goal of the STC is to automatically classify incoming short texts into different categories, thereby preventing users from being overwhelmed by the massive amount of raw web data. Furthermore, STC can be readily applied to a wide range of natural language processing (NLP) tasks, such as sentiment analysis, dialogue systems, and offensive language detection.

In the earlier stage, Latent Semantic Analysis (LSA) (Dumais 2004) and its extensions, such as Independent Component Analysis (ICA) (Comon 1994) and Language Independent Semantic (LIS) kernel (Kim et al. 2014), play an important role in the STC. These approaches have the capability to extract potential semantic structures while classifying short texts by combining matrix decomposition techniques with machine learning-based classification algorithms, including Naïve Bayes, K-nearest neighbors, and support vector machine (Song et al. 2014). However, it is worth noting that these approaches are computationally expensive and heavily reliant on feature engineering.

Subsequently, STC methods based on the deep neural network (DNN) have garnered considerable attention due to

✉ Hongfei Lin
hflin@dlut.edu.cn

Changrong Min
mcr19940816@gmail.com

Yonghe Chu
yonghechu@163.com

Bolin Wang
wangbolin@mail.dlut.edu.cn

Liang Yang
liang@dlut.edu.cn

Bo Xu
xubo@dlut.edu.cn

1 Criminal Investigation Police University of China, No.83 Tawan St, Shenyang, Liaoning, China

2 School of Computer Science and Technology, Dalian University of Technology, No. 2 Linggong Road, Dalian 116024, Liaoning, China

3 School of Information Science and Engineering, Henan University of Technology, No. 100 Lianhua Street, Zhengzhou 450001, Henan, China

the advancements in deep learning in recent years. These methods primarily employ Convolutional Neural Network (CNN), Recurrent Neural Networks (RNN), and other neural network structures (Mirończuk and Protasiewicz 2018) as the backbone. The CNN is effective in extracting local features, such as N-gram features, while the RNN captures long-distance features from texts. However, despite their individual strengths in prioritizing locality and sequentiality, both CNN and RNN overlook the valuable global word co-occurrence information that encompasses non-consecutive and long-distance semantics.

More recently, the GCN has emerged as a promising approach for addressing the STC task (Linmei et al. 2019; Zhang et al. 2020; Liu et al. 2020). For example, Yao et al. (2019) treat the text classification task as node classification, where they construct a text graph consisting of word and text nodes. They then employ a GCN to learn the node embeddings via message passing and predict the labels of text nodes. Wu et al. (2012) construct a word-level graph for each document, connecting nodes within a fixed-size window. This approach enables better capture of local features and significantly reduces memory consumption. Linmei et al. (2019) propose a Heterogeneous Graph Attention Network that incorporates a double-layer attention mechanism for text classification. By utilizing a heterogeneous information network, this method can integrate various types of additional information and the relationships between them.

However, it is worth noting that the aforementioned GCN-based approaches primarily focus on texts of normal length, and few studies have investigated their effectiveness on short texts. Moreover, applying the GCN to short texts poses a significant challenges. Firstly, short texts are semantically sparse and lack sufficient context (Song et al. 2014).This sparsity issue results in the absence of connections between word pairs that are highly correlated in our common sense. Secondly, most GCN-based methods rely solely on SoftMax or Cross-Entropy objective functions to learn an optimal representation of a given text that is most similar to its ground-truth label. These methods ignore the intra-class and inter-class geometrical structures in the global semantic space, resulting in unclear classification boundaries among samples from different categories.

To address the aforementioned challenges, we propose a novel GCN-based STC method named **To**pic-aware **Co**sine **G**raph **C**onvolutional **N**etwork (**ToCo-GCN**), which effectively mitigates the sparsity problem and fully utilizes the global geometric structures of short texts. Specifically, given an STC corpus, the ToCo-GCN first captures its latent topic distributions of words and short texts. Meanwhile, a text graph that takes the words and short texts as nodes is constructed. Then, the ToCo-GCN regards the latent topics as virtual nodes and construct a topic-aware text graph. Based on the topic prior, this graph directly connects word pairs within each topic cluster, alleviating the sparsity of the text graph. During the graph learning stage, to learn discriminative text embeddings, the ToCo-GCN captures the intra-class and inter-class geometric structures over the graph in a cosine space. Specifically, inspired by the literature (Wang et al. 2018), the ToCo-GCN utilizes the cosine value of the angle between text embeddings and label embeddings to measure both the inter-class and intra-class geometric structures. Minimizing such geometric constraint enforces angular between short texts from the same category to be smaller while angular between short texts from different categories to be larger in the cosine space. It makes short texts of the same category more compact in space while pushing short texts from different categories farther away. By doing this, the discriminative boundaries between different categories of short texts are now clearer, which effectively enhances task performance. The contributions of our work are summarized as follows:

- We propose the ToCo-GCN, which fully exploits geometric structures of data by simultaneously considering intra-class and inter-class geometric structures in the STC. Additionally, we make use of topic information to alleviate the sparsity problem for better adapting the model to short texts.
- We experimentally evaluate the ToCo-GCN with other state-of-the-art models on 8 STC datasets. The ToCo-GCN shows significant improvements in terms of Accuracy and Macro-F1 score compared to the baselines.

The remainder of the paper is organized as follows: In section 2, related work on the STC is introduced. In Sect. 3, we introduce the ToCo-GCN in detail. The experimental results and analyses are given in Sect. 4. Finally, we conclude this paper in Sect. 5.

2 Related work

In this paper, we revise the existing researches on the STC task from two perspectives: traditional STC methods and deep learning-based STC methods.

2.1 Traditional STC methods

Earlier studies on short text classification mainly made use of statistical machine-learning techniques. For instance, a bag-of-words (BoW) model built with rare vocabulary information is proposed in the literature (Heap et al. 2017). Samant et al. (2019) classify short texts based on the Vector Space Model with a new weighting mechanism for each word. Moreover, other feature models, such as TFIDF and n-grams, are also employed for short text classification (Yang et al. 2021; Cavnar et al. 1994). However, both the BoW and the

VSM do not well solve the high-dimensionality and sparsity problems inherent in short texts. Feature selection methods involving the Chi-square test (CHI), GINI index (GINI), and dictionary learning are proposed to address the high-dimensionality problem (Liu et al. 2022). For solving the sparsity problem, Li et al. (2017) enrich short text features by using concepts from an external corpus Probase [17]. Alsmadi et al. [18] make use of a keyword expansion method to extend the feature space of short texts. Though these approaches improve the problems and perform better than previous work, their performances still have a gap with deep learning-based methods.

2.2 Deep learning-based STC methods

With the breakthrough of deep learning in the past few years, more and more text classification approaches employ deep neural networks to automatically learn semantic features and classify texts. For example, Kim (2014) proposes a CNN-based model with multi-channel to classify texts. Zhang et al. (2015) propose character-level CNN that models different levels of features, improving the accuracy of text classification. Directly applying these frameworks will perform poorly because the above-mentioned problems of short texts are ignored by them. Then, Hu et al. (2018) leverage a combination of the CNN and Support Vector Machine to enhance the performance of short texts. Moreover, Alam et al. (2020) represent short texts with words and entities and exploit a CNN-based model to classify short texts. To obtain better short text features, Yin et al. (2019) make use of the attention mechanism on the character level and incorporate it into a CNN-based model. In addition to these CNN-based methods, Recurrent Neural Network and its variants are also widely explored in short text classification (Lee and Dernoncourt 2016; Liu and Guo 2019). However, both the CNN-based and RNN-based methods fail to make use of global word co-occurrence information in a corpus that carries non-consecutive and long-distance semantics.

More recently, Graph Neural Networks (Zhou et al. 2020), which concentrate on coping with arbitrary non-Euclidean spatial data, have been well exploited in text classification. In addition to the aforementioned textGCN and the TL-GNN (Huang et al. 2019), Zhang et al. (2020) propose the TextING that encodes each document as a single graph and inductively learns node embeddings with a double-layer GNN. Moreover, Liu et al. [?] propose a tensor graph that is merged by semantic, syntactic, and sequential graphs of a corpus. Different from these methods, Ding et al. (2020) propose the HyperGAT that involves word-word edges. However, these methods will not perform well for short texts because of lacking context information. Thus, GCN-based models for short texts are proposed. For example, Linmei et al. (2019) propose the HGAT that simultaneously models topics, entities, and documents. The entities are associated with knowledge graphs. Ye et al. (2020) propose the STGCN, which develops a corpus-level graph based on not only traditional text relations but also topic relations, alleviating the sparseness of short texts. However, these GCN-based approaches for the STC task fail to consider both intra-class and inter-class geometric structures of samples in a corpus. This impedes models from learning text representations that are representative as well as discriminative.

3 Methodology

3.1 Problem definition

We now formulate the task of STC, whose training dataset contains N labeled samples $\mathcal{D} = \{(x_i, \mathbf{y}_i)\}_{i=1}^{N}$. The notations x and $\mathbf{y} \in \{0, 1\}^C$ denote the raw short text and category label, respectively. The goal of our work is to train a GCN-based classifier over $\mathcal{D}$, enabling to distinguish the category of a given short text.

3.2 The basic GCN

In this subsection, we introduce the basic GCN that operates directly on graph-structured data. Specifically, given a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$. The notion $\mathcal{V} = \{v_1, v_2, \ldots, v_{\mathrm{T}}\}$ denote the set of nodes, while the $\mathcal{E}$ denotes the set of edges. T is the total number of nodes in the graph $\mathcal{G}$. We use $\mathbf{U} = [u_1, u_2, \ldots, u_{\mathrm{T}}] \in \mathbb{R}^{\mathrm{T}\times \mathrm{d}}$ to denote the node features, where d is the dimension of node features. The corresponding adjacent matrix is denoted as $\mathbf{A} \in \{0, 1\}^{\mathrm{T}\times \mathrm{T}}$, where 1/0 denotes the component corresponds to an edge or not. Besides, each node of the two graphs is with self-loop. The degree matrix $\mathbf{D}$ is a diagonal matrix and $\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij}$. Then, for a single-layer GCN, the node features can be updated by the following equation:

$$\mathbf{L}^{(1)} = \rho\left(\tilde{\mathbf{A}}\mathbf{U}\mathbf{W}_0\right) \tag{1}$$

where $\mathbf{L}^{(1)} \in \mathbb{R}^{\mathrm{T}\times \mathrm{k}}$ is the learned node feature matrix. k is the expected dimension of node features. $\tilde{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$ is the normalized symmetric adjacency matrix of the $\mathbf{A}$. $\mathbf{W}_0$ is trainable parameters of the GCN. ρ is the activation function, such as ReLU. By doing this, the single-layer GCN can induce node features from the neighbors via first-order message-passing mechanism, learning structure-aware node features.

Therefore, a multi-layer GCN can bring information from higher-order neighborhoods. The learning procedure of node

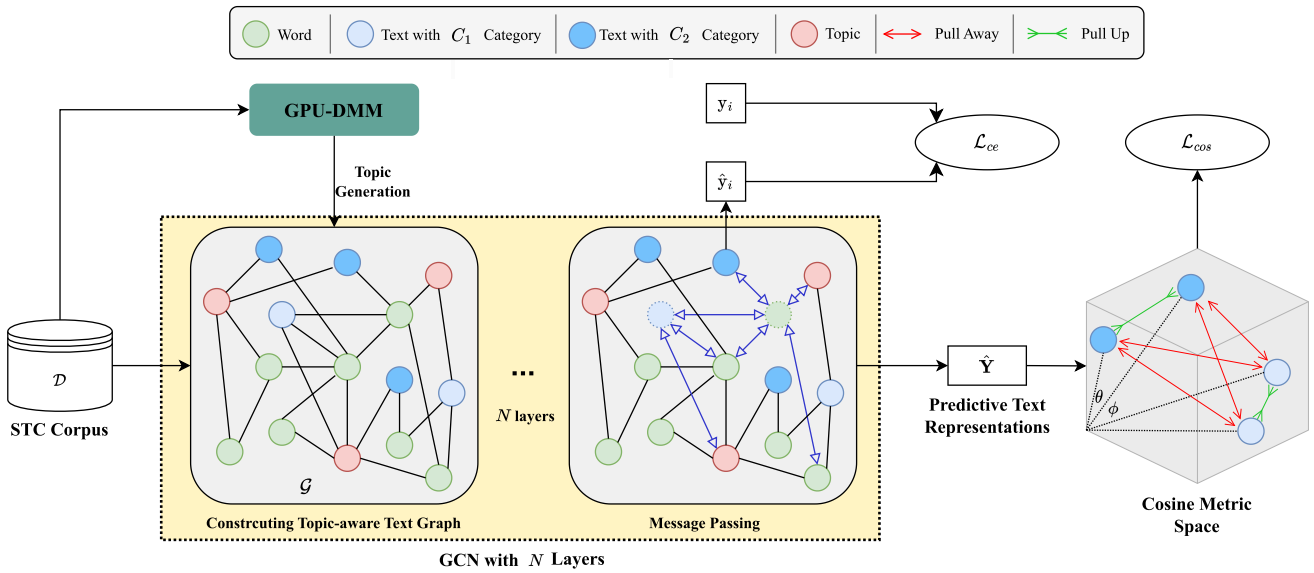

Fig. 1 The architecture of the ToCo-GCN. This method first generates topic distribution for the incoming STC corpus $\mathcal{D}$ via the GPU-DMM and then constructs a topic-aware text graph $\mathcal{G}_s$. Then, a N-layer GCN is employed to learn the node embeddings. Eventually, such predictive results of samples are leveraged to calculate the total loss

features can be further formulated as:

$$\mathbf{L}^{(j+1)} = \rho\left(\tilde{\mathbf{A}}\mathbf{L}^{(j)}\mathbf{W}_j\right) \tag{2}$$

where j denotes the number of layers. $\mathbf{W}_j$ is trainable parameters of the j-th layer.

3.3 The proposed ToCo-GCN

In this subsection, we introduce the structures and training objective of the proposed ToCo-GCN. The overall framework is shown in the Fig. 1.

3.3.1 Constructing a topic-aware text graph

Given the corpus $\mathcal{D}$, the ToCo-GCN first constructs a text graph $\mathcal{G}_s = \{\mathcal{V}_s, \mathcal{E}_s\}$. The set of nodes $\mathcal{V}_s = \left\{v_1^s, v_2^s, \ldots, v_{\mathrm{T}_s}^s\right\}$ consists of two parts: words and texts, where the T_s denotes the total number of nodes in the graph $\mathcal{G}_s$. The set of edges $\mathcal{E}_s$ also contains two kinds of relations: word-to-word and word-to-text. The former is defined by the Point-wise Mutual Information (PMI) values, while the latter is defined by the TFIDF values (Yao et al. 2019). The PMI value of a given word pair $< v_i^s, v_j^s >$ is calculated as:

$$\mathrm{PMI}(v_i^s, v_j^s) = \log \frac{p(v_i^s, v_j^s)}{p(v_i^s)p(v_j^s)} \tag{3}$$

$$p(v_i^s, v_j^s) = \frac{\#Count(v_i^s, v_j^s)}{N_w} \tag{4}$$

$$p(v_i^s) = \frac{\#Count(v_i^s)}{N_w} \tag{5}$$

where N_w denotes the total number of word nodes. $\#Count(v_i^s, v_j^s)$ is the co-occurrence frequency of the word pair in a corpus. However, for short texts, some synonyms or highly related word pairs do not co-occur in the window due to the sparsity problem. Hence, the $p(v_i^s, v_j^s)$ will equal zero. The PMI value of these word pairs will be an Infinitesimal. The quality of node representations might be degraded due to the message-passing between the node pairs is unavailable in the first layer of the GCN.

To improve the sparsity of short texts, we enrich the text graph with topic information that provides latent connections between words and documents. We leverage the topic model GPU-DMM (Li et al. 2016) that derives topic distributions of short texts and distributions of words under each topic. The latent topics are as nodes in text graph. Moreover, topic-document edges and word-topic edges are constructed. Then, the adjacent matrix $\mathbf{A}^{\mathbf{s}}$ the graph $\mathcal{G}$ can be defined as follows:

$$\mathbf{A}^{\mathbf{s}}{}_{ij} \begin{cases} \mathrm{PMI}(i,j) & i,j \text{ are words, } \mathrm{PMI}(i,j) > 0 \\ \mathrm{TFIDF}_{ij} & i \text{ is a text, } j \text{ is a word} \\ \mathbf{R}_{ij}^{(tw)} & i \text{ is topic, } j \text{ is word} \\ \mathbf{R}_{ij}^{(tx)} & i \text{ is topic, } j \text{ is text} \\ 1 & \text{self-loop} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where $\mathbf{R}_{ij}^{(tw)}$ denotes the extra word-topic relation. It equals to 1 when the j-th word is associated with the i-th topic. $\mathbf{R}_{ij}^{(tx)}$

is the topic-text relation. It is initialized by the maximum probability of the topic distribution of the i-th document. Similar to word and text nodes, latent topic nodes are also initialized with one-hot vectors. Hence, the node embedding matrix $\mathbf{X} \in \mathbb{R}^{\mathrm{T_s} \times \mathrm{k}}$ can be initialized by an identity matrix $\mathbf{I}$.

3.3.2 Updating node embeddings over the graph

After obtaining the adjacent matrix $\mathbf{A^s}$ and the node embeddings $\mathbf{X}$, we employ a two-layer GCN to learn node embeddings over the topic-aware text graph $\mathcal{G}_s$. The learning process can be formulated as follows:

$$\mathbf{Z}^{(0)} = \mathrm{ReLU}\left(\tilde{\mathbf{A}}^{\mathbf{s}}\mathbf{X}\mathbf{W}_0\right) \tag{7}$$

$$\mathbf{Z}^{(1)} = \mathrm{SoftMax}\left(\tilde{\mathbf{A}}^{\mathbf{s}}\mathbf{Z}^{(0)}\mathbf{W}_1\right) \tag{8}$$

where $\mathbf{W}_0$ and $\mathbf{W}_1$ are the parameters of the first layer and the second layer, respectively. $\mathbf{Z}^{(1)} \in \mathbb{R}^{\mathrm{T_s} \times C}$ denotes the node embeddings derived from the last GCN layer. Such a two-layer structure allows node to pass messages from second-order neighborhood over the graph. The ReLU and SoftMax are the activation functions.

3.3.3 Optimizing with cosine-based training objective

For optimizing the ToCo-GCN, we design a cosine-based objective function $\mathcal{L}_{total}$ that fully considers global geometric structures of short texts in the semantic space. The $\mathcal{L}_{total}$ is formulated as:

$$\mathcal{L} = \mathcal{L}_{ce} + \lambda * \mathcal{L}_{cos} \tag{9}$$

where the first term $\mathcal{L}_{ce}$ is implemented by the cross-entropy that enforces to learn features close to the ground-truth labels. The second term $\mathcal{L}_{cos}$ is a cosine-margin loss that models the intra-class and inter-class geometric structures of short texts in a cosine space. λ is a trade-off parameter that balances the two terms.

Given the predictive results of texts $\mathbf{Z_d} = \{\mathbf{z}_i\}_{i=1}^{N} \subset \mathbf{Z}^{(1)}$, the cross-entropy term $\mathcal{L}_{ce}$ is calculated as follows:

$$\mathcal{L}_{ce} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C}\mathbf{y}_{ij}\log\left(z_{ij}\right) \tag{10}$$

where C is the number of classes. $\mathbf{y}_{ij}$ equals 1 when the j-th label is true of the i-th text, otherwise it equals 0. Minimizing the $\mathcal{L}_{ce}$ allows the ToCo-GCN to learn representative features of short texts.

The second regularization term $\mathcal{L}_{cos}$ is leveraged to construct both intra-class and inter-class geometric structures in cosine space. It is calculated asfollows:

$$\mathcal{L}_{cos} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s\left(\cos\left(\theta_{\mathbf{y}_i z_i}\right)-m\right)}}{e^{s\left(\cos\left(\theta_{\mathbf{y}_i, z_i}\right)-m\right)} + \sum_{j\neq\mathbf{y}_i} e^{s\cos\left(\theta_{j,z_i}\right)}} \tag{11}$$

where $\mathrm{m} \geq 0$ is a cosine margin that can better improve the ability of discriminative. $\theta_{\mathbf{y}_i,z_i}$ denotes the angle between the i-th text and its corresponding label $\mathbf{y}_i$ in the angular space, while θ_{j,z_i} represents the angle of the i-th text to the other labels. The ToCo-GCN simultaneously minimize the intra-class compactness and maximize the inter-class separation in cosine space. When minimizing the $\mathcal{L}_{\cos}$, the angle $\theta_{\mathbf{y}_i,z_i}$ between the text d_i and the weight vector of its ground-truth label $\mathbf{y}_i$ will be minimized, and the angle $\theta_{j,i}$ between z_i and the weight vector of the j-th category, where j represents any label other than $\mathbf{y}_i$, will be maximized. The $\cos\left(\theta_{j,z_i}\right)$ is calculated by:

$$\cos\left(\theta_{j,z_i}\right) = \frac{\mathrm{q}_j^T}{\left\|\mathrm{q}_j^T\right\|}\frac{z_i}{\|z_i\|} \tag{12}$$

where the q_j denotes the weight vector of the j-th category. Moreover, we use the L_2 normalization term to remove radial variations.

4 Experiments

In this section, we first introduce several publicly available short text datasets and experimental details. Then, we introduce some sate-of-the-art baselines for comparison. Finally, the experimental results and analysis are provided.

4.1 Experimental settings

4.1.1 Datasets

We evaluate the performance of our method on the following 8 benchmarks:

(1) R8: This dataset represents a subset of the Reuters 21578 dataset.
(2) CR: This dataset is a customer product review dataset.
(3) MR: This dataset is a movie review dataset.
(4) SST-binary (SST-Bi): This dataset is the Stanford Sentiment Treebank dataset.
(5) StackOverflow (STOW): This dataset includes selected questions and the corresponding labels posted on stackoverflow.com from July 31, 2012, to August 14, 2012.
(6) Biomedical (BIO):Biomedical is a subset of the challenge data published on the BioASQ's website, where 19974 paper titles from 20 groups are randomly selected.

Table 1 The statistics of the STC datasets

Datasets	#docs	#tokens	#entities	#train	#test	#classes
R8	7,674	7,688	15,362	5,485	2,189	8
MR	10,662	18,764	29,426	7,108	3,554	2
CR	3,773	7,683	11,456	2,515	1,258	2
SST-Bi	10,754	6,972	17,726	8,544	2,210	5
TagMyNews	32,549	38,629	71,178	26,040	6,509	7
BIO	19,974	28,753	48,727	17,976	1,998	20
Electronics	188,626	291,804	480,430	150,900	37,726	796
STOW	20,000	32,639	52,639	16,000	4,000	20

(7) TagMyNews: This dataset consists of titles of English news from really simple syndication feeds.
(8) Electronics (Tayal et al. 2019, 2020): This dataset is collected from Amazon e-commerce platform.

The detailed statistics of each dataset are shown in the Table 1.

4.1.2 Training details

We follow the pre-processing of the textGCN to clean and tokenize texts. We remove non-English characters, the stop words, and low-frequency words appearing less than 5 times for seven datasets other than MR. For the MR dataset, since the texts are too short, all words have remained after the cleaning and tokenizing operations. Table 1 demonstrates the statistics of the datasets, including the number of documents, the number of average tokens and entities, the number of classes, and the proportion of texts containing entities in parentheses. For the ToCo-GCN, the embedding dimension of the first GCN layer is set to 200, while the window size is 20. We set the learning rate as 0.001, and the dropout rate is set as 0.5. The value of the epoch is set to a maximum of 1,000 with an early stopping mechanism. Moreover, we make use of Adam as the optimizer following the literature (Alam et al. 2020). For baselines that leverage pre-trained word embeddings as input, we make use of 300-dimensional GloVe word embeddings [1] (Pennington et al. 2014). We evaluate the classification performance using test accuracy (denote as Acc in short) and macro-averaged F1 score (denote as F1 in short).

4.1.3 Baselines

To evaluate the effectiveness of the proposed ToCo-GCN, we select the following 10 well-performed STC methods as baselines:

(1) TFIDF + LR: This method uses the TFIDF as the feature of short texts and takes the Logistics Regression as the classifier.
(2) textCNN: This method is based on the Convolutional Neural Network (Kim 2014). We develop two variants of the textCNN: CNN_{rand} and CNN_{nsta}, respectively. The former randomly initializes word embeddings, while the latter uses the pre-trained word embeddings.
(3) LSTM: We develop two LSTM variants: $\text{LSTM}_{\text{rand}}$ and $\text{LSTM}_{\text{nsta}}$, respectively.
(4) PV-DBOW: This method uses a paragraph vector model (Le and Mikolov 2014) as the text features and takes the Logistic Regression as the classifier.
(5) FastText (Joulin et al. 2016): This method treats the average of word/n-grams embeddings as document embeddings and feeds such document embeddings into a linear classifier.
(6) SWEM (Shen et al. 2018): The method applies pooling strategies over pre-trained word embeddings.
(7) LEAM (Wang et al. 2018): This method considers the label information, which jointly learns word and label embeddings. The label information is implemented via the textual label description.
(8) textGCN: This method forms an STC corpus into a text graph with both document and word nodes and jointly learns node representations via message passing over the graph.
(9) TL-GNN: This method treats each document as a single graph and employs GCN to learn its representation.
(10) TG-Transformer (Zhang and Zhang 2020): This method a novel Transformer-based heterogeneous graph neural network, which is a large-sized corpus and ignores the heterogeneity of the text graph.

4.2 Results and analysis

We evaluate the proposed ToCo-GCN over 8 datasets for the STC task. The results are respectively shown in Figs. 2 and 3. From the results, we can draw the following observations:

[1] http://nlp.stanford.edu/data/glove.6B.zip

Table 2 The experimental results of all comparing methods in terms of Accuracy (Acc) and Macro-F1 (F1). The best results are represented in **bold**. The second-best results are underlined

Datasets	R8		CR		MR		SST-Bi	
metric	Acc	F1	Acc	F1	Acc	F1	Acc	F1
TFIDF+LR	93.7	91.2	62.0	60.7	74.5	74.2	58.9	58.7
CNN_{rand}	94.0	92.0	71.1	68.4	75.0	74.6	62.0	61.5
CNN_{nsta}	95.7	93.6	76.2	74.2	77.8	77.7	66.7	66.4
$LSTM_{rand}$	93.7	91.4	63.8	60.9	75.1	74.7	67.3	67.0
$LSTM_{nsta}$	96.1	93.3	63.7	60.7	77.3	76.5	70.2	69.9
PV-DBOW	85.9	84.6	65.3	63.8	61.1	60.9	66.0	65.6
FastText	83.0	82.5	75.0	72.6	67.6	66.8	69.8	69.8
SWEM	95.3	91.7	72.9	70.4	76.7	75.9	65.9	65.7
LEAM	93.3	93.0	74.3	72.5	77.0	77.0	65.0	64.2
textGCN	97.1	96.3	73.8	72.0	76.7	76.2	70.8	70.4
TL-GNN	97.8	96.8	74.0	73.4	74.3	73.9	71.0	70.9
TG-Transformer	97.4	96.2	73.5	72.7	75.1	74.6	69.1	69.0
ToCo-GCN	97.9	97.7	76.4	76.0	78.2	77.9	73.8	73.7
Ablation Study								
w/o Topic	97.2 ↓	96.3 ↓	75.0 ↓	74.8 ↓	76.9 ↓	76.8 ↓	72.9 ↓	72.7 ↓
w/o $\mathcal{L}_{cos}$	96.1 ↓	95.4 ↓	74.5 ↓	74.0 ↓	78.8 ↑	78.4 ↑	72.0 ↓	71.7 ↓
5-fold Cross Validation (Average Results)								
textGCN	96.3	95.0	75.1	74.9	77.5	76.4	68.8	68.7
ToCo-GCN	98.1	97.4	78.2	78.0	80.4	80.5	72.4	72.2

Table 3 The experimental results of all comparing methods in terms of Accuracy (Acc) and Macro-F1 (F1). The best results are represented in **bold**. The second-best results are underlined

Datasets	TagMyNews		BIO		Electronics		STOW	
metric	Acc	F1	Acc	F1	Acc	F1	Acc	F1
TFIDF+LR	45.7	41.4	64.5	61.1	60.8	59.0	83.5	84.2
CNN_{rand}	42.6	37.0	62.6	60.7	56.3	52.9	85.6	86.2
CNN_{nsta}	46.9	40.5	65.9	63.3	59.7	57.5	88.3	88.8
$LSTM_{rand}$	42.1	38.3	63.7	59.9	59.7	56.4	85.2	84.9
$LSTM_{nsta}$	57.5	52.0	67.6	64.8	62.1	59.9	87.1	87.0
PV-DBOW	48.8	43.1	63.4	60.9	59.5	56.6	87.4	88.0
FastText	67.5	62.0	66.7	65.0	62.7	60.8	86.1	86.3
SWEM	64.8	59.6	65.5	64.0	63.5	59.4	85.5	85.0
LEAM	68.4	62.9	65.9	63.4	62.3	60.7	84.7	84.4
textGCN	78.0	73.9	67.2	65.3	66.6	64.6	87.9	88.4
TL-GNN	77.9	73.7	68.8	66.5	66.5	64.7	88.6	89.4
TG-Transformer	78.2	73.9	69.0	68.2	**67.2**	**66.0**	88.9	89.4
ToCo-GCN	**79.5**	**75.3**	**69.7**	**68.5**	**67.2**	65.8	**90.4**	**90.3**
Ablation Study								
w/o Topic	79.2 ↓	74.7 ↓	67.9 ↓	66.0 ↓	66.4 ↓	64.9 ↓	89.2 ↓	89.2 ↓
w/o $\mathcal{L}_{cos}$	78.7 ↓	74.2 ↓	69.2 ↓	68.3 ↓	66.7 ↓	65.4 ↓	88.4 ↓	88.7 ↓
5-fold Cross Validation (Average Results)								
textGCN	77.1	72.0	64.3	64.7	65.5	63.9	86.2	86.4
ToCo-GCN	78.4	73.7	66.5	66.0	66.3	64.5	88.7	89.4

(1) Overall, the proposed ToCo-GCN outperforms all the baselines by a large margin in terms of Acc and F1 score. For example, the ToCo-GCN achieves increases of 2.8% in Acc and 2.8% in F1 score on the SST-Bi dataset. This indicates that introducing the topic information of short texts and the cosine margin-based loss function can benefit the STC task.
(2) However, the ToCo-GCN shows a slight decrease of 0.2% in F1 score on the Electronics dataset. One possible reason is that the scale of this dataset is too large, and the TG-Transformer has many more parameters than the ToCo-GCN. Therefore, the TG-Transformer has a better ability to learn high-quality short text representations.
(3) We observe that the graph neural network (GNN)-induced methods (textGCN, TL-GNN, TG-Transformer, and the ToCo-GCN) achieve better performances than the non-GNN-induced methods in terms of Acc and F1 score on most benchmarks. This indicates that treating the corpus as a whole graph and globally learning word as well as text representations over the graph is efficient for the STC task.
(4) We observe that STC methods with pre-trained word embeddings, such as $\text{LSTM}_{\text{nsta}}$ and CNN_{nsta}, continuously outperforms those with randomly initialized word embeddings. This indicates that pre-trained word embeddings provide rich semantic information that can benefit the STC task.
(5) Moreover, we observe that the PV-DBOW method, which ignores the word order, performs poorly on most datasets. This indicates that word orders are important to capture latent semantics of short texts.

4.3 Ablation study

We further evaluate the effectiveness of the two main components of the ToCo-GCN: the topic information and the cosine margin-based loss function $\mathcal{L}_{cos}$. The ablative results are respectively shown in Figs. 2 and 3. From the results, we observe that when either the topic information is removed

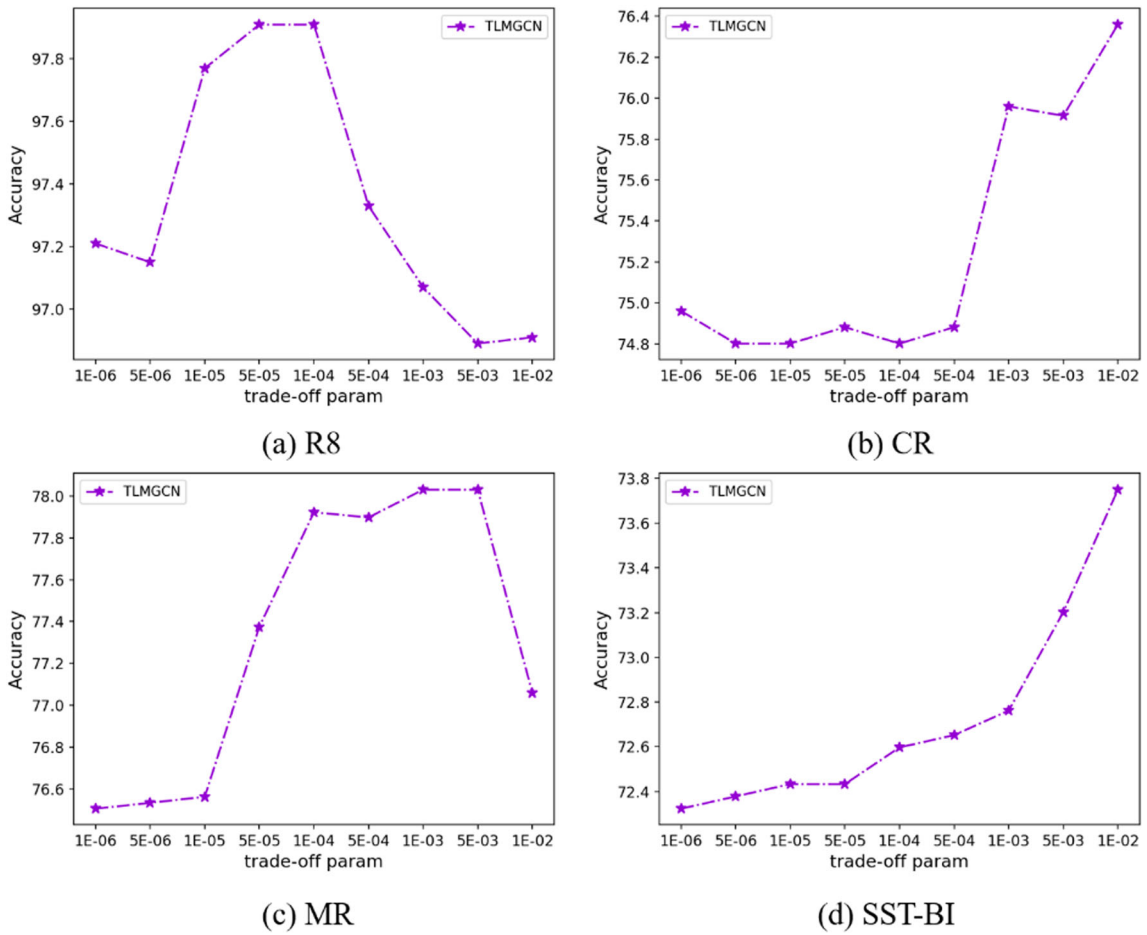

Fig. 2 The performance of the ToCo-GCN in terms of Acc under different values of the trade-off parameter λ

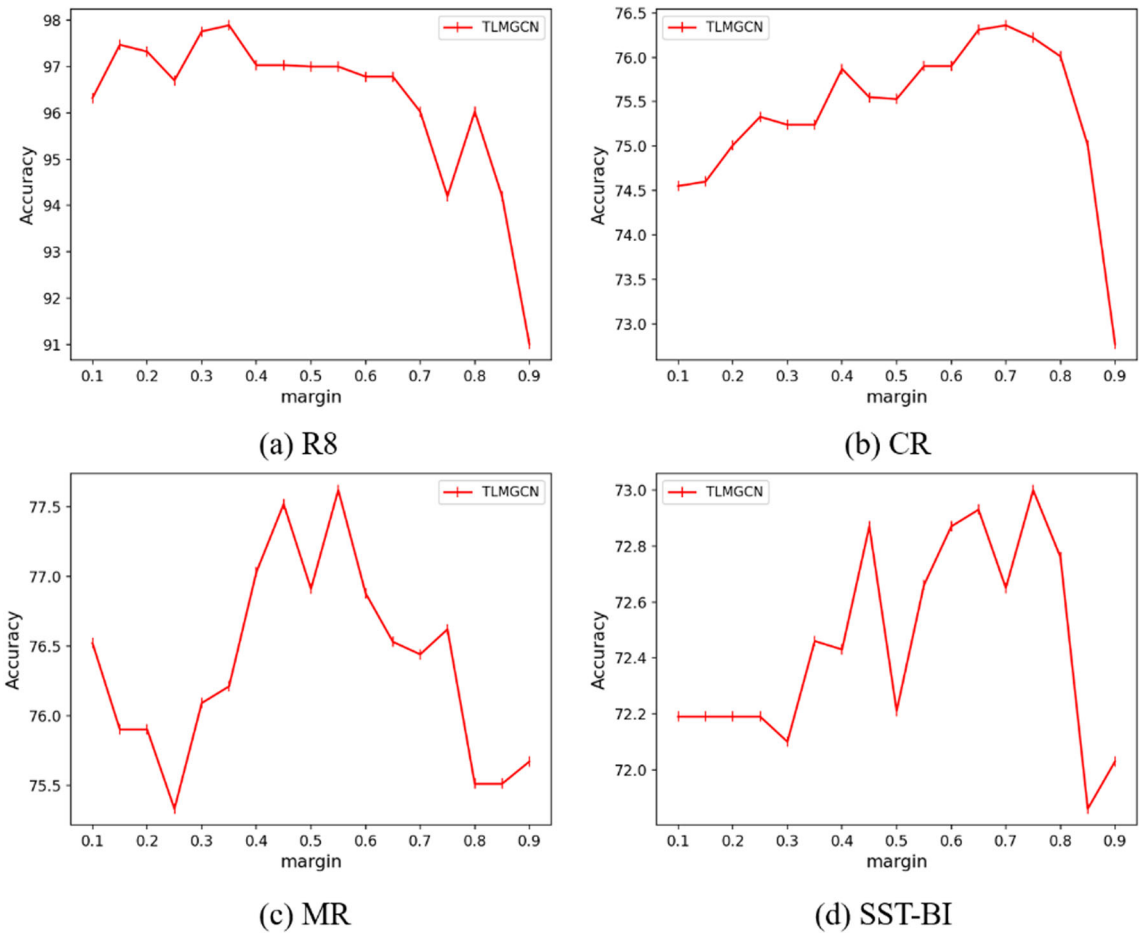

Fig. 3 The performance of the ToCo-GCN in terms of Acc under different values of the margin m

from the text graph or the $\mathcal{L}_{cos}$ is removed, the ToCo-GCN's performance in terms of accuracy and F1 significantly decreases over most datasets. This indicates that introducing the topic information can efficiently shorten the semantic interaction distances between words or words and documents over the graph, improving the quality of text representations. However, we also observe that the ToCo-GCN shows increases of 0.6% and 0.5% in terms of accuracy and F1 on the MR dataset after removing the $\mathcal{L}_{cos}$. One possible reason for this is that the angle between some text pairs that do not belong to the same category is incorrectly minimized, while the angle between some pairs that belong to the same category is maximized.

4.4 Parameter sensitivity

We further explore the efficiency of several important parameters of the ToCo-GCN: the trade-off parameter λ, the cosine margin m, the number of latent topics, and the dimension of embeddings, respectively.

4.4.1 Effect of the trade-off parameter λ

We evaluate the effectiveness of the parameter λ, which controls the importance of $\mathcal{L}_{cos}$. The value of λ is in the range of $[10^{-6}, 10^{-2}]$. Figure 2 demonstrates the variation of accuracy with the increase of λ. Based on the results, we draw the following observations:

(1) On the R8 and MR datasets, the performance of the ToCo-GCN generally shows a trend of initially increasing and then decreasing. When $\lambda = 10^{-4}$, the ToCo-GCN achieves the optimal result on the R8 dataset, while for the MR dataset, the optimal value is $\lambda = 5 \times 10^{-3}$. The reason for this may be that samples with different categories in the R8 dataset always leverage specific words or phrases to describe the news. Therefore, these samples can be well classified by the ToCo-GCN when the discriminative constraint $\mathcal{L}_{cos}$ is set to a small value. However, the MR dataset focuses on sentiment classification, and some samples may simultaneously contain both positive and negative sentiment expressions, which

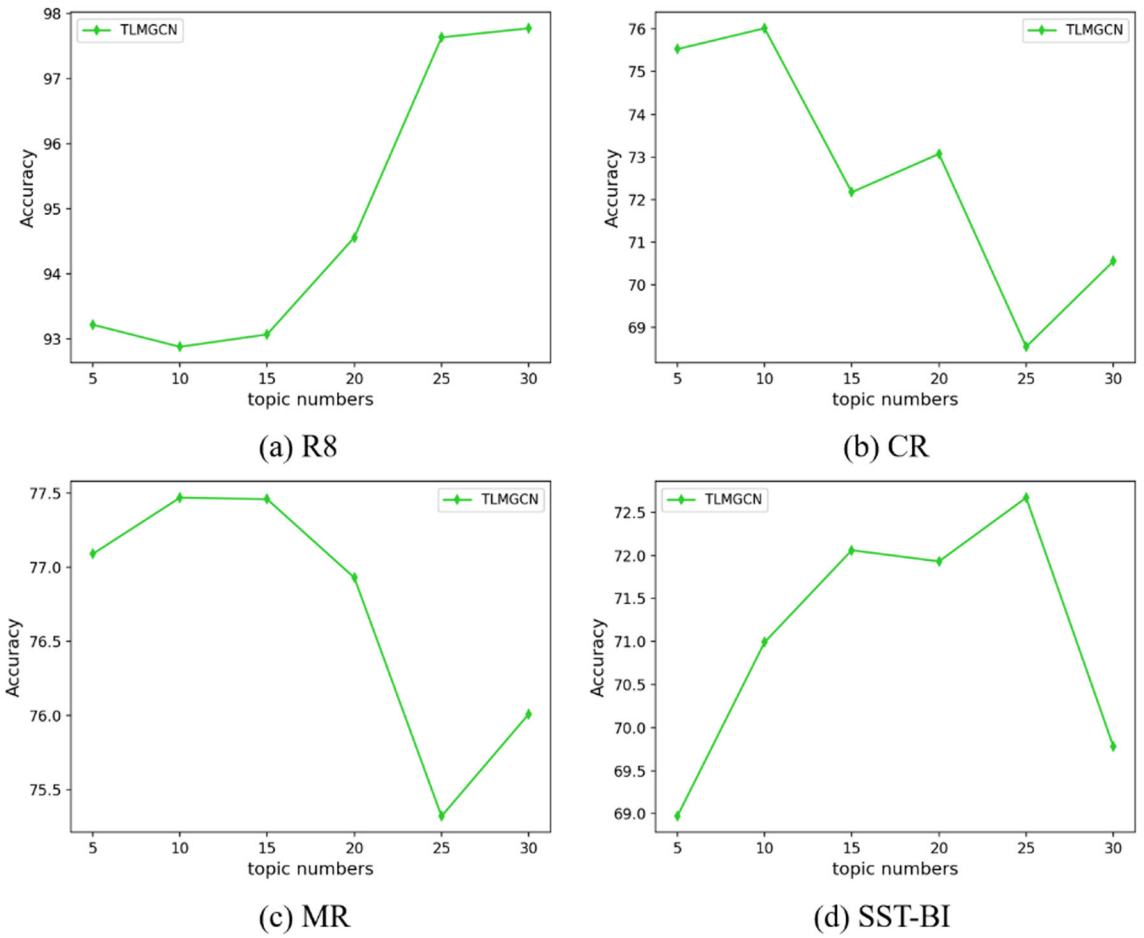

Fig. 4 The performance of the ToCo-GCN in terms of Acc under different numbers of topics

are difficult to distinguish even for human beings. Therefore, a larger value of $\mathcal{L}_{cos}$ is needed to enforce the ToCo-GCN to learn discriminative sentiment-specific features for the MR dataset.

(2) In contrast to the above performances, the performance of the ToCo-GCN on the CR and SST-Bi datasets gradually improves as the value of λ increases, and the ToCo-GCN performs best when $\lambda = 10^{-2}$ on both datasets. This indicates that only using the cross-entropy loss $\mathcal{L}_{ce}$ to minimize the difference between individual sample predictions and ground-truth labels is insufficient on the CR and SST-Bi datasets. Therefore, the ToCo-GCN further utilizes the global information of samples in the cosine space to learn discriminative text features, effectively improving the task performance of STC.

4.4.2 Effect of the cosine margin

We evaluate the effectiveness of the parameter m, which controls the anger between sample-pairs in the cosine space. The value of m is in the range of [0.1, 0.9]. Figure 3 shows the variation of accuracy with the increase of m. Based on the results, we draw the following observations:

(1) On the R8 and CR datasets, the performance of the ToCo-GCN first gradually increases to a peak and then rapidly decreases within the [0.8, 0.9] range. This upward trend indicates that the ToCo-GCN can learn discriminative text features while sufficiently preserving specific semantic information for each text. However, the rapid decline may be due to the excessively large margin m incorrectly enforcing some samples from different categories to be closer.

(2) Compared to the performances on the above two datasets, the performances of the ToCo-GCN on the MR and SST-Bi datasets are more sensitive to changes in the value of m. The possible reason for this is that the distinction between samples from different categories is relatively low, resulting in less clear category decision boundaries in the cosine semantic space. Therefore, even small changes in the value of m can have a noticeable impact on the task performances.

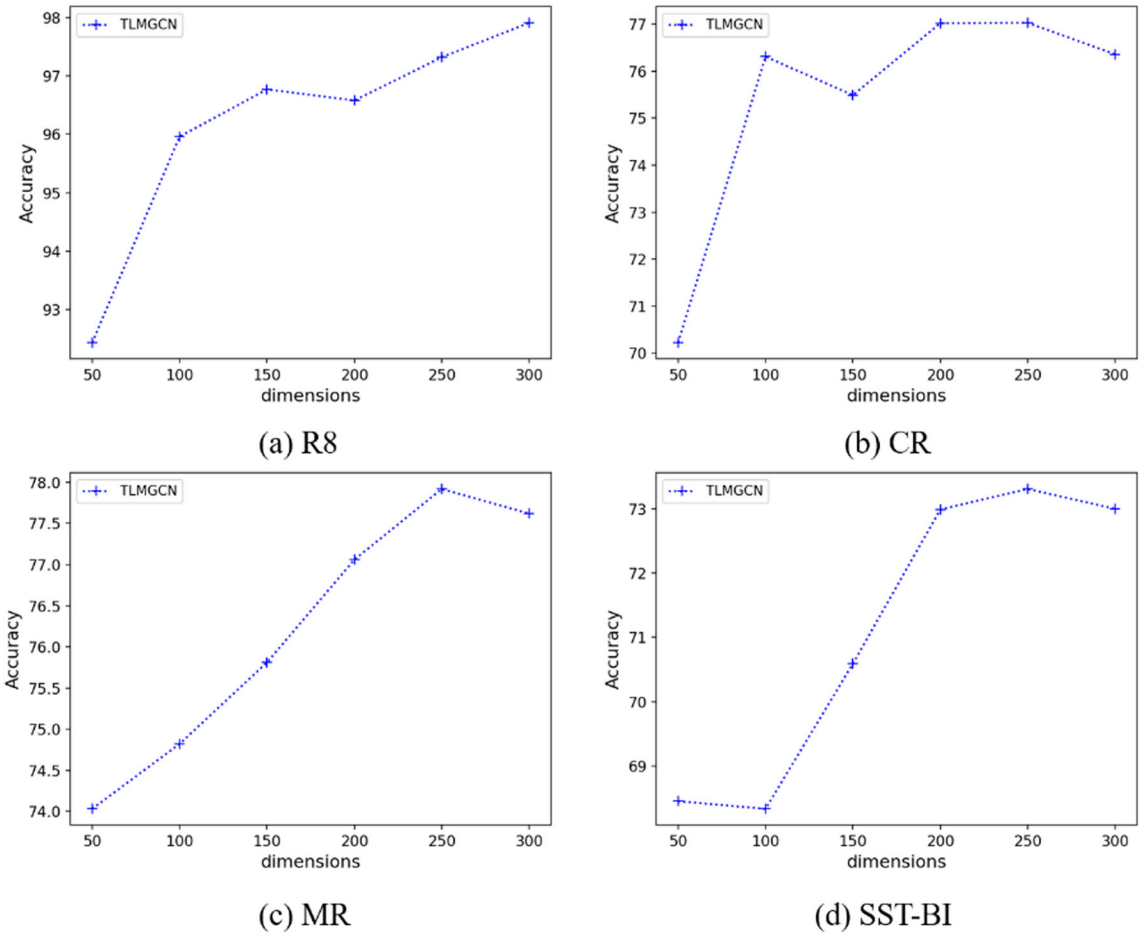

Fig. 5 The performance of the ToCo-GCN in terms of Acc under different dimensions

4.4.3 Effect of the latent topics

We further analyze the impact of the number of latent topics on the performance of the ToCo-GCN across four datasets. The results are shown in Fig. 4. Overall, the performance of the ToCo-GCN varies across the four datasets, and the optimal performance on the CR, MR, and SST-Bi datasets corresponds to 10, 15, and 25 topic nodes, respectively. This suggests that appropriately introducing topic nodes can reduce the distance between semantically related but distant word pairs or word-document pairs over the text graph, effectively improving the efficiency of capturing global semantic information. However, we observe that the ToCo-GCN performs best when the number of topic nodes is set to 30 on the R8 dataset. This may be because the R8 dataset has more categories than the other three datasets, and therefore, more fine-grained topic information allows the ToCo-GCN to better capture discriminative information between different categories.

4.4.4 Effect of the embedding dimensions

We evaluate the impact of different embedding dimensions in the 1^{st} GCN layer on the performance of the ToCo-GCN. The results are reported in Fig. 5. From the results, we observe that the ToCo-GCN achieves optimal results on the CR, MR, and SST-Bi datasets when the dimension is set to 250. Additionally, on these three datasets, the performance initially increases and then slowly decreases as the dimension increases. This indicates that as the dimension increases, the ToCo-GCN can capture more discriminative and rich semantics. However, excessively large dimensions may introduce unnecessary noise and hurt the performance of the STC task.

4.5 Visualization of classification results

Figure 6 demonstrates the t-SNE (Van der Maaten and Hinton 2008) visualization of the first layer text embeddings learned from the R8 dataset. With the increase of m, samples of the **acq** class and samples of the **earn** class can maintain good intra-class aggregation as well as inter-class separation. The

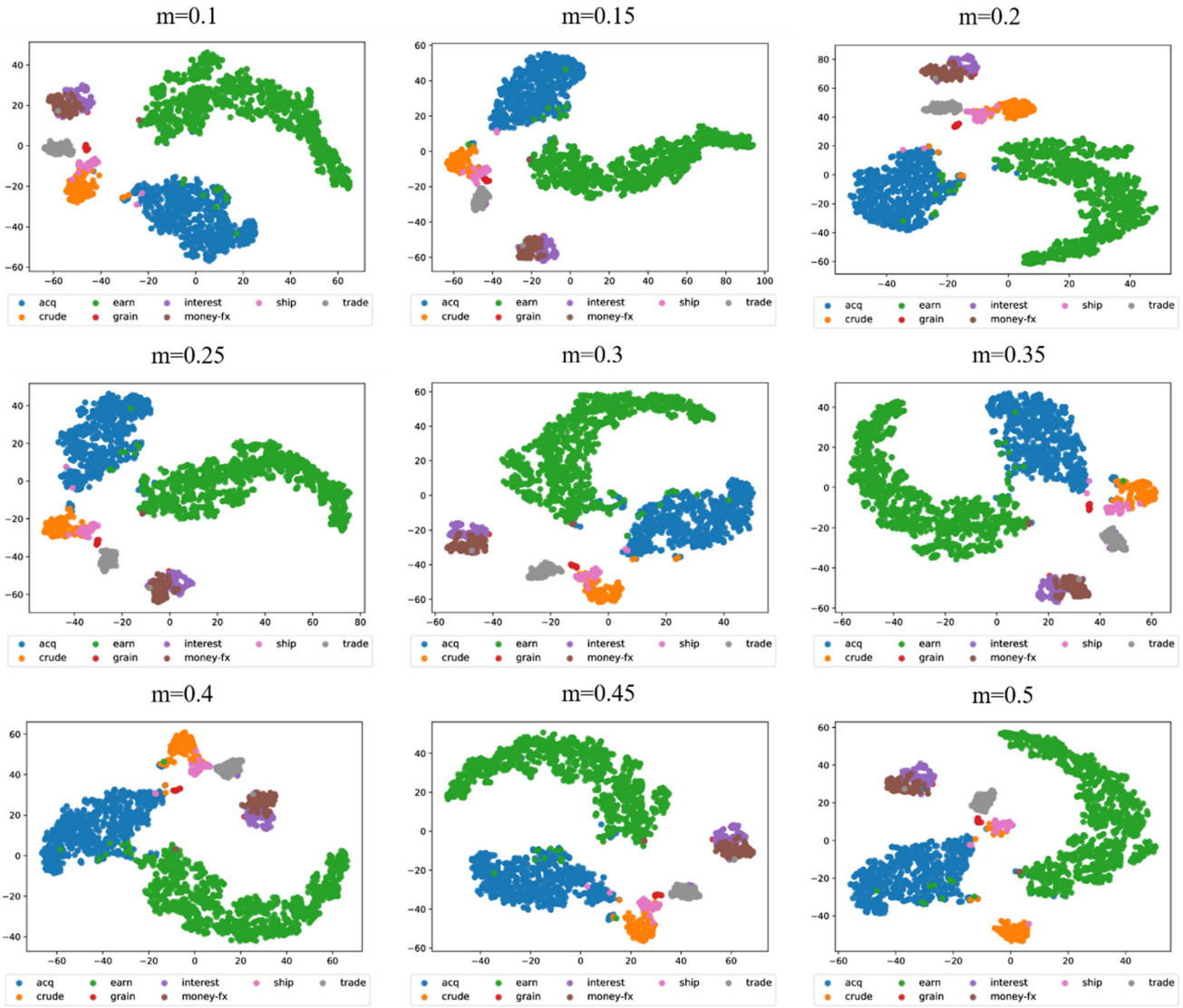

Fig. 6 The t-SNE visualization of text embeddings obtained by the ToCo-GCN on the R8 dataset

reason is that the number of samples of the two categories is larger compared to the other classes, hence our model is able to learn discriminative features even with smaller margins. However, for categories with only a few samples, we can observe that the boundary between category A and other categories gradually increases as the margin increases from 0.1 to 0.35. Additionally, there is an overlap between the **interest** class and the **money-fx** class, and this issue only slightly improves as m increases from 0.1 to 0.5. We believe there are two reasons for this: firstly, the two classes are similar in terms of topics or content, and secondly, the limited number of samples hinders the model from learning distinctive features of the two classes.

4.6 Time consumption of model training and testing

We further compared the proposed ToCo-GCN with the textGCN in terms of time consumption during training and testing stages, as shown in Table 4. From the results, we can observe that there is almost no significant difference in the time consumption per training epoch between the ToCo-GCN and textGCN. This indicates that introducing topic information and the discriminative constraint $\mathcal{L}_{cos}$ into the ToCo-GCN may not impose a heavy computational burden. However, on the MR dataset, the overall training time of the ToCo-GCN (4.3s) is significantly longer than that of textGCN (3.1s). This may be due to optimizing with the $\mathcal{L}_{cos}$ slows down the convergence speed of the ToCo-GCN. Therefore, under the early stopping mechanism, the ToCo-GCN requires more training epochs to achieve fitting.

Table 4 Comparison of average time consumption (in seconds) on 10 runs. The running environment is on the NVIDIA A100 80 G GPU

Methods	ToCo-GCN			textGCN		
Mode	Training		Testing	Training		Testing
Datasets	All	Epoch	–	All	Epoch	–
R8	6.8483	0.0410	0.0142	6.4375	0.0392	0.0144
CR	1.7368	0.0089	0.0034	1.6283	0.0087	0.0031
MR	4.3074	0.1320	0.0483	3.1052	0.1307	0.0473
SST-Bi	1.8196	0.0132	0.0063	1.7382	0.0125	0.0055
TagMyNews	34.5042	1.7536	0.5842	28.7183	1.6805	0.5609
BIO	24.5903	1.3082	0.2875	23.5570	1.2414	0.2903
Electronics	278.3964	15.7500	6.4088	259.6590	14.7367	6.0248
STOW	30.8663	1.1503	0.3389	28.9036	1.0740	0.3224

5 Conclusion and future work

Although the GCN-based methods in text classification construct graphs at the text level, which contains both local co-occurrence relations and global co-occurrence relations, and makes use of multi-layer GCN to exploit the two relations in the raw corpus to learn text embeddings based on pre-trained embeddings, they do not fully employ geometric structures of labeled data. In this paper, we propose a novel method for short text classification, called Topic-aware Cosine Graph Convolutional Neural Network (ToCo-GCN). The ToCo-GCN cannot only learn representative text embeddings but also can make use of underlying intra-class and inter-class geometric structures to enhance the power of discriminative. Experiments on four benchmark data sets show that the proposed model is superior to the GCN and several competing existing short text classification methods. In the future, we will investigate how to further extend the graph neural networks to other NLP downstream tasks, as well as how to leverage external knowledge to enhance the ability of graph learning to capture task-relevant features from a global perspective.

Author Contributions All authors contributed to the experiments and writings. Material preparation, data collection and analysis were performed by Changrong Min, Yonghe Chu and Bolin Wang. The first draft of the manuscript was written by Changrong Min and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding This work was supported by the National Natural Science Foundation of China (Grant numbers No.62076046).

Data availability Data openly available in a public repository.

Declarations

Conflict of interest The authors certify that there is no conflict of interest with any individual/organization for the present work.

Informed consent Informed consent was obtained from all individual participants included in the study.

Ethical approval This article does not contain any studies with human participants performed by any of the authors.

References

Alam M, Bie Q, Türker R, Sack H (2020) Entity-based short text classification using convolutional neural networks. In: Knowledge Engineering and Knowledge Management: 22nd International Conference, EKAW 2020, Bolzano, Italy, September 16–20, 2020, Proceedings 22, pp. 136–146. Springer

Cavnar WB, Trenkle JM, et al (1994) N-gram-based text categorization. In: Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval, vol. 161175, p. 14. Las Vegas, NV

Comon P (1994) Independent component analysis, a new concept? Signal processing 36(3):287–314

Ding K, Wang J, Li J, Li D, Liu H (2020) Be more with less: Hypergraph attention networks for inductive text classification. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 4927–4936

Dumais ST (2004) Latent semantic analysis. Annual Review of Information Science and Technology (ARIST) 38:189–230

Heap B, Bain M, Wobcke W, Krzywicki A, Schmeidl S (2017) Word vector enrichment of low frequency words in the bag-of-words model for short text multi-class classification problems. arXiv preprint arXiv:1709.05778

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 (EMNLP-IJCNLP), pp. 3444–3450

Hu Y, Li Y, Yang T, Pan Q (2018) Short text classification with a convolutional neural networks based method. In: 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV), pp. 1432–1435. IEEE

Joulin A, Grave E, Bojanowski P, Douze M, Jégou H, Mikolov T (2016) Fasttext. zip: Compressing text classification models. arXiv preprint arXiv:1612.03651

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

Kim K (2014) Chung, B-s, Choi, Y, Lee, S, Jung, J-Y, Park, J: Language independent semantic kernels for short-text classification. Expert Systems with Applications 41(2):735–743

Lee JY, Dernoncourt F (2016) Sequential short-text classification with recurrent and convolutional neural networks. In: Proceedings of NAACL-HLT, pp. 515–520

Le Q, Mikolov T (2014) Distributed representations of sentences and documents. In: International Conference on Machine Learning, pp. 1188–1196. PMLR

Li P, He L, Wang H, Hu X, Zhang Y, Li L, Wu X (2017) Learning from short text streams with topic drifts. IEEE transactions on cybernetics 48(9):2697–2711

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 (EMNLP-IJCNLP), pp. 4821–4830

Liu G, Guo J (2019) Bidirectional lstm with attention mechanism and convolutional layer for text classification. Neurocomputing 337:325–338

Liu Y, Li P, Hu X (2022) Combining context-relevant features with multi-stage attention network for short text classification. Computer Speech & Language 71:101268

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, pp. 8409–8416

Li C, Wang H, Zhang Z, Sun A, Ma Z (2016) Topic modeling for short texts with auxiliary word embeddings. In: Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 165–174

Maaten L, Hinton G (2008) Visualizing data using t-sne. Journal of machine learning research **9**(11)

Mirończuk MM, Protasiewicz J (2018) A recent overview of the state-of-the-art elements of text classification. Expert Systems with Applications 106:36–54

Pennington J, Socher R, Manning CD (2014) Glove: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543

Samant SS, Murthy NB, Malapati A (2019) Improving term weighting schemes for short text classification in vector space model. IEEE Access 7:166578–166592

Shen D, Wang G, Wang W, Min MR, Su Q, Zhang Y, Li C, Henao R, Carin L (2018) Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 440–450

Song G, Ye Y, Du X, Huang X, Bie S (2014) Short text classification: a survey. Journal of multimedia **9**(5)

Tayal K, Nikhil R, Agarwal S, Subbian K (2019) Short text classification using graph convolutional network. In: NIPS Workshop on Graph Representation Learning

Tayal K, Rao N, Agarwal S, Jia X, Subbian K, Kumar V (2020) Regularized graph convolutional networks for short text classification. In: Proceedings of the 28th International Conference on Computational Linguistics: Industry Track, pp. 236–242

Wang G, Li C, Wang W, Zhang Y, Shen D, Zhang X, Henao R, Carin L (2018) Joint embedding of words and labels for text classification. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 2321–2331

Wang H, Wang Y, Zhou Z, Ji X, Gong D, Zhou J, Li Z, Liu W (2018) Cosface: Large margin cosine loss for deep face recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5265–5274

Wu W, Li H, Wang H, Zhu KQ (2012) Probase: A probabilistic taxonomy for text understanding. In: Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, pp. 481–492

Yang T, Hu L, Shi C, Ji H, Li X, Nie L (2021) Hgat: Heterogeneous graph attention networks for semi-supervised short text classification. ACM Transactions on Information Systems (TOIS) 39(3):1–29

Yao L, Mao C, Luo Y (2019) Graph convolutional networks for text classification. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 7370–7377

Ye Z, Jiang G, Liu Y, Li Z, Yuan J (2020) Document and word representations generated by graph convolutional network and bert for short text classification. In: ECAI 2020, pp. 2275–2281. IOS Press, ???

Yin F, Yao Z, Liu J (2019) Character-level attention convolutional neural networks for short-text classification. In: Human Centered Computing: 5th International Conference, HCC 2019, Čačak, Serbia, August 5–7, 2019, Revised Selected Papers 5, pp. 560–567. Springer

Zhang Y, Yu X, Cui Z, Wu S, Wen Z, Wang L (2020) Every document owns its structure: Inductive text classification via graph neural networks. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 334–339

Zhang H, Zhang J (2020) Text graph transformer for document classification. In: Conference on Empirical Methods in Natural Language Processing (EMNLP)

Zhang X, Zhao J, LeCun Y (2015) Character-level convolutional networks for text classification. Advances in neural information processing systems **28**

Zhou J, Cui G, Hu S, Zhang Z, Yang C, Liu Z, Wang L, Li C, Sun M (2020) Graph neural networks: A review of methods and applications. AI open 1:57–81

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.