applied sciences

MDPI

Article

Heterogeneous Graph Neural Network for Short Text Classification

Bingjie Zhang, Qing He * and Damin Zhang

College of Big Data & Information Engineering, Guizhou University, Guiyang 550025, China
* Correspondence: qhe@gzu.edu.cn

Abstract: Aiming at the sparsity of short text features, lack of context, and the inability of word embedding and external knowledge bases to supplement short text information, this paper proposes a text, word and POS tag-based graph convolutional network (TWPGCN) performs short text classification. This paper builds a T-W graph of text and words, a W-W graph of words and words, and a W-P graph of words and POS tags, and uses Graph Convolutional Network (GCN) to learn its feature and performs feature fusion. TWPGCN only focuses on the structural information of text graph, and does not require pre-training word embedding as initial node features, which improves classification accuracy, increases computational efficiency, and reduces computational difficulty. Experimental results show that TWPGCN outperforms state-of-the-art models on five publicly available benchmark datasets. The TWPGCN model is suitable for short text or ultra-short text, and the composition method in the model can also be extended to more fields.

Keywords: heterogeneous graph; graph neural network; short text

check for updates

Citation: Zhang, B.; He, Q.; Zhang, D. Heterogeneous Graph Neural Network for Short Text Classification. *Appl. Sci.* **2022**, *12*, 8711. https://doi.org/10.3390/app12178711

Academic Editor: Valentino Santucci

Received: 20 July 2022
Accepted: 28 August 2022
Published: 30 August 2022

1. Introduction

Short text usually refers to the text form with a relatively short length, generally no more than 160 characters. Short text classification is a kind of text mining technology and an important research direction in natural language processing (NLP). With the continuous development of NLP, people's requirements for text mining are not limited to ordinary texts, and short texts including literature summaries, news public opinions, and opinion comments have begun to enter the public's field of vision [1]. Analysis of short texts has important application value, such as opinion mining for user comments, topic detection for social media, and public opinion warning [2].

Different from long texts, short texts have the characteristics of sparse sample features, lack of context, and many noisy data [3]. They make the semantics difficult to distinguish, and the comprehension deviation cannot be eliminated, and the results are a huge difference between the semantics of the short text and the results learned by the model. Differences make traditional text classification algorithms not well suited for short texts. In order to solve such problems, relevant scholars have used traditional text classification methods, such as SVM [4], BAYES [5], KNN [6], to achieve short text classification by reducing its discreteness reducing noise features. However, these methods originate from traditional text classification. Through some methods to improve the classification accuracy, there are not the most suitable methods for short text classification.

Benefiting from the characteristics of high randomness of Graph Neural Network (GNN) structure and unique node update method, it can make up for the shortcomings of traditional text classification methods in short texts. In recent years, some scholars have begun to study GNN for short text classification [7–9]. GNN is suitable for data with limited labels and lack of features. Adding edges to defined nodes can better reduce noise, focus on existing features. However, the number of such studies is limited, and the problems faced by short text classification have not been completely solved. Therefore, the research of GNN suitable for short text is very meaningful.

This research also faces challenges. Most of the existing short text classification based on GNN rely on external knowledge supplements, such as word vectors or external knowledge bases. Such methods achieve the effect of knowledge supplementation at the expense of increasing the amount of computation, but doing so is not necessarily the best. First, due to different scenarios, pre-trained word vectors may not necessarily improve the effect of text classification, but increase the difficulty of mapping [10]. Second, excessive reliance on external knowledge bases may bring more noise and reduce model accuracy. For example, in the case of relying on an external knowledge base, the word "apple" in the two texts "Jobs founded Apple" and "Newton thought for a long time under the apple tree" will be given the same meaning in the knowledge base link. Since short texts do not provide much information, such misleading knowledge can lead to fatal results, and we need a model that focuses more on the characteristics of the text itself.

In this paper, we propose a heterogeneous graph convolutional network suitable for short texts. This network does not rely on word vectors or external knowledge bases. It emphasizes the relationship between nodes themselves, and achieves strong classification performance at low computational cost. In summary, the contributions of this paper are as follows:

1. We propose a heterogeneous graph around text, words, and POS tags, using a multi-layer GCN to learn the features of the three graphs separately, and combine the information of different graphs to learn short texts.
2. We do not rely on pre-trained word vectors or external knowledge bases, but only rely on the co-occurrence relationship between words and the TF-IDF relationship between words and text to extract features, which reduces the difficulty of network learning.
3. Extensive experiments on five benchmark datasets show that our model outperforms state-of-the-art models compared to several GNN model-based short text classification methods.

2. Related Work

Traditional text classification methods are K-nearest neighbors (KNN) [11], which classifies an unlabeled sample by finding the class with the most samples over the k closest labeled samples. Decisions tree (DT) is a supervised tree structure induction classification algorithm. It uses the attributes of the samples as nodes and the values of the attributes as the branches of the tree structure. Commonly used decision tree algorithms are ID5 [12], CART [13], C4.5 [14] and so on. Wang et al. [15] proposed to perform context word embedding through word windows of different sizes to calculate the vector representation of multi-scale semantic units in short texts, and then select word representations that are close enough in the semantic units to form a text expansion matrix, and finally pass CNN is used for classification, which reduces the discrete type of short text features to achieve better learning results. Huang et al. [16] aimed at the expansion of the KNN short text classification algorithm, which led to the problem of reducing the efficiency of short text classification, using chi-square statistics to extract training samples in a category that are more similar to the characteristics of the category, and split the training space into finer details. To improve the quality of training samples, the number of texts compared by the KNN short text classification algorithm reduce, thereby improving the efficiency. ZHANG et al. [17] proposed a method of using an external database containing the text to be classified, using LDA to train a topic model, extracting text topics, and then integrating the topics into the text to achieve text expansion. Using the expanded text training, the resulting vector is classified, so that the error rate is significantly reduced. Kettaf et al. [18] proposed a combination of a neuronal network, discriminant analysis, SVM, and other methods to identify the author of a given text. This method presents the text as elements of a separable Hilbert space, keep a lot of stylistic information and not requiring a prior reduction of the dimension, enabling special text classification. Henrique et al. [19] proposed a network that considers the semantic similarity between paragraphs to improve the text classification task. This method can be combined with traditional networks and is a highly adaptable improvement method.

Graph convolutional neural network is a kind of GNN. Bruna et al. [20] first proposed the concept of GCN in 2014. Graph convolution semi-supervised learning uses convolution operations to combine the feature vectors of nodes with the graph structure between nodes. Every time the feature vectors of nodes undergo a graph convolution operation, they update their own feature vectors with adjacent nodes through the graph structure, so that similar nodes have similar feature vectors [21]. Therefore, GCN have attracted widespread attention, and related scholars introduced them into many fields, such as entity recognition [22], image understanding [23], action recognition [24], and text classification [25].

Graph neural networks for short text classification have received more and more attention from scholars, and many models have been proposed based on them. Yao et al. [7] proposed a method to classify short texts based on a heterogeneous graph between words and texts. It puts words and texts in a large global graph, and proposed two kinds of edges to connect them, and learns textual information through an overall GCN. This is one of the first articles to propose such a method, and many subsequent studies are based on it. In order to make up for the lack of short text features, such as the difficulty of capturing and lacking context, Zhang et al. [8] proposed a graph convolutional neural network to classify text using attachment relationships and supplementing information with external knowledge bases. The external knowledge base is really helpful for short text classification, it adds more features to the text, giving the model more initial knowledge, but it also brings more noise. In order to solve this problem, Wang et al. [9] proposed a method called SHINE to construct three graphs for words, entities extracted from external knowledge bases, and labels corresponding to words, and use the GCN model to learn its characteristics and classification. Although this method also uses an external knowledge base, it only uses the external knowledge base to select entities in the model without adding new knowledge, which effectively reduces the noise caused by referencing the external knowledge base.

3. Methodology

3.1. Construction of the Graph

The method proposed in this paper constructs a heterogeneous graph around text, words, and POS tags to which words belong. The T-W graph between text and words is shown in Figure 1a, which includes text nodes and word nodes, and links between text nodes and word nodes. The edge is characterized by the term frequency–inverse relationship between words and text document frequency term frequency–inverse document frequency (TF-IDF) value, it is a statistical method to evaluate the importance of words to a text set or a piece of text in a corpus. The importance of a word increases proportionally to the number of times it occurs in the text, but at the same time decreases inversely to the frequency it occurs in the corpus. The TF-IDF value is calculated as follows:

$$\mathrm{TF}-\mathrm{IDF}_{T-W} = \frac{n}{N} \times log\frac{D}{1+d} \tag{1}$$

where n is the number of occurrences of the word in the text, N is the sum of the occurrences of all words in the text, D is the total number of texts in the corpus, and d is the number of documents that contain the word.

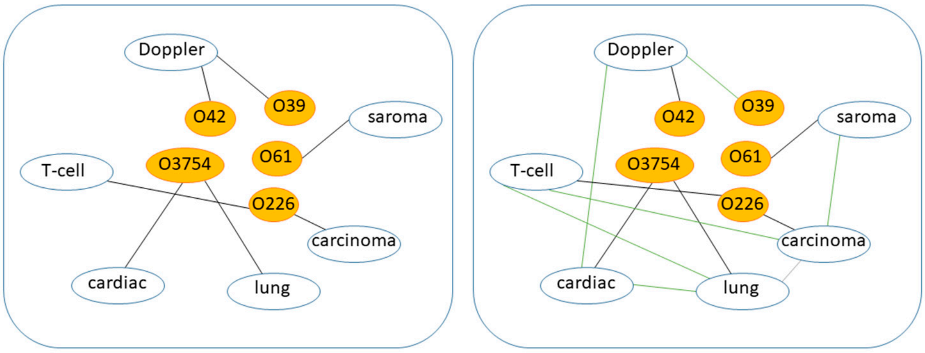
The W-W graph between words and words is shown in Figure 1b, which contains text nodes and word nodes, and the text nodes are connected to the word nodes they contain. The feature of the edge is Pointwise Mutual Information (PMI) value. The PMI value can solve the misleading calculation of high-frequency words. If a word co-occurs with many words, its weight will be reduced. Conversely, if a word only co-occurs with individual words, its weight is increased. This will make the fixed collocations in the text more recognizable, and make common words less recognizable, reducing the impact on the text. The co-occurrence method of words is obtained by window sliding in the text, and the calculation method of PMI value is as follows:

$$\mathrm{PMI}(i,j) = log\frac{p(i,j)}{p(i)p(j)} \tag{2}$$

$$\mathrm{p(i,j)} = \frac{\#W(i,j)}{\#W} \tag{3}$$

$$\mathrm{p(i)} = \frac{\#W(i)}{\#W} \tag{4}$$

where #*W*(*i*,*j*) refers to the number of sliding windows that nodes *i* and *j* appear together, #*W(i)* refers to the number of sliding windows that node *i* appears, and #*W* is the total number of sliding windows. A positive PMI indicates that the words co-occur, that is, they will appear together; a negative PMI indicates that the words do not appear together. This paper only builds edges between word nodes with a positive PMI.



(**a**) Graph T-W (**b**) Graph W-W

Figure 1. Yellow nodes represent text nodes, white nodes represent word nodes, black lines represent edges between text and words, and green lines represent edges between words and words.

The W-P diagram between words and POS tag is shown in Figure 2. Since a word plays different components in a sentence. For example, in the sentence "can you open a can?", the first can is a modal verb, and the second can are nouns, and they belong to different POS tags, so a word may have one or several POS tags. In our constructed W-P graph, there are text nodes, word nodes, and POS tag nodes. The text node is connected to the word node it contains, the feature of the edge is 1, the connection between the word node and the POS tag node is also formed, and the feature of the edge is also the IDF value between the word and the POS tag. The calculation method is as follows:

$$\mathrm{IDF}_{T-P} = log\frac{P}{1+p} \tag{5}$$

where *P* is the total number of POS tags in the corpus and *p* is the number of POS tag for the word.

The heterogeneous graph constructed according to words, texts and POS tags is completed. Different from the homogeneous graph, the heterogeneous graph contains more abundant information. Through the information dissemination of different types of edges, POS tag nodes can help the network filter more important word nodes, and finally summarize the information into text nodes. Enlarging the characteristics of the text itself, focusing on the text through TF-IDF, and reducing noise, such a composition can give full play to its advantages in the classification of pictures and short texts.

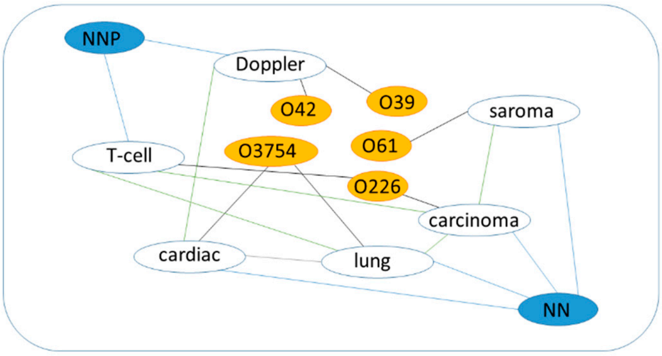


Figure 2. In Graph W-P, blue nodes represent POS tag nodes, blue lines represent edges connecting words to POS tag.

3.2. Graph Convolutional Networks

For a given undirected graph $G = (v, \varepsilon, A,)$ where ν, ε, $A \in R^{C\times C}$ are the node set, edge set and symmetric adjacency matrix of the graph, respectively, use GCN to input Learning the feature $H_l \in R^{C\times d}$ and the adjacency matrix $A \in R^{C\times C}$, we can get.

$$H_{l+1} = \delta\left(\hat{A} H_l W_l\right) \tag{6}$$

$$\hat{A} = \hat{\mathrm{D}}^{-\frac{1}{2}} \hat{A} \hat{\mathrm{D}}^{-\frac{1}{2}} \tag{7}$$

$$\hat{A} = A + I \tag{8}$$

Among them: H_{l+1} is the feature output of single-layer GCN, which is the adjacency matrix after normalization, $W_l \in R^{C\times d}$ is the weight matrix, and $\hat{A}$ is the undirected graph with self-connected adjacency before normalization Matrix, I is a unit diagonal matrix, is a degree matrix in the form of a diagonal matrix, and $\delta(\cdot)$ is a nonlinear activation function. In the graph neural network proposed in this paper, the activation function adopts the ReLU function, and the formula is as follows:

$$\mathrm{ReLU}(\mathrm{x}) = \begin{cases} 0 & x <= 0 \\ x & x > 0 \end{cases} \tag{9}$$

3.3. TWPGCN

The TWPGCN proposed in this paper is shown in Figure 3. The text is automatically processed to form a heterogeneous graph containing text, words, and POS tags. Two-layer GCN is performed on the three subgraphs in the heterogeneous graph, and the results obtained each time are $\hat{x}^i_{T-W}$, $\hat{x}^i_{W-W}$, $\hat{x}^i_{W-P}$, which can be regarded as is to interpret each short text in terms of documents, words, and POS tags. Finally, the short text features we get can be expressed as

$$x^i = \hat{x}^i_{T-W} \parallel \hat{x}^i_{W-W} \parallel \hat{x}^i_{W-P} \tag{10}$$

Among them, $\parallel$ represents feature splicing, and the feature representation of the text in the i-th layer is obtained after splicing the features learned from the subgraph. After the double-layer GCN network, the features will be classified and predicted by the Sotfmax layer. The Sotfmax calculation is as follows:

$$\mathrm{Softmax}(z_i) = \frac{e^{z_i}}{\sum_{C=1}^{C} e^{z_C}} \tag{11}$$

Among them, z_i is the output value of the i-th node, and C is the number of output nodes, that is, the number of categories of classification. The output value of the multi-classification can be converted into a probability distribution in the range [0, 1] and 1 through the Softmax function.

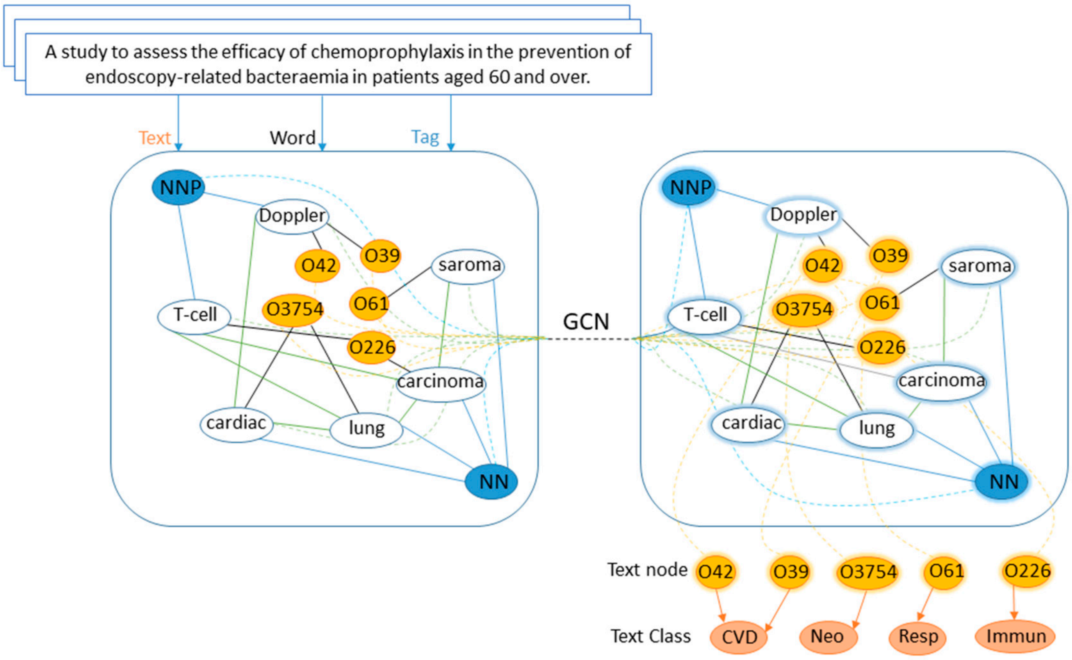

A study to assess the efficacy of chemoprophylaxis in the prevention of endoscopy-related bacteraemia in patients aged 60 and over.
Text
Word
Tag
NNP
Doppler
O39
O42
saroma
O61
T-cell
O3754
O226
carcinoma
cardiac
lung
NN
GCN
Text node
O42
O39
O3754
O61
O226
Text Class
CVD
Neo
Resp
Immun

Figure 3. TWPGCN The model learns and classifies heterogeneous graphs composed of short texts.

4. Experiment

4.1. Datasets

We use five public datasets to train the model. The details of the datasets are shown in Table 1:

i. **20NG**: A dataset collected by Ken Lang for news text classification [26].
ii. **R8** and **R52**: 8-category dataset (R8) and 52-category dataset (R52) for Reuters news.
iii. **Ohsumed**: a subset of the bibliographic Ohsumed dataset [27] used in [28] where the title is taken as the short text to classify.
iv. **MR**: a movie review dataset for sentiment analysis [29].

Table 1. Summary of short text datasets used.

	#Text	**#Classed**	**#Words**	**#POS Tags**
20NG	18,846	20	40,760	40
R8	7674	8	7688	36
R52	9100	52	8892	36
Ohsumed	7400	23	11,764	37
MR	10,662	2	18,764	38

POS tags in graph are extracted by Python's NLTK library. Overall, the 20NG dataset has a larger volume, while the R8 dataset has a smaller volume, which may have an impact on the accuracy of model classification. The number of POS tags averaged 38, while the categories of the dataset ranged from 2 to 52.

4.2. Baseline Model

Baseline models can be divided into the following categories:

a. Traditional model: **TF-IDF+LR** (bag-of-words model with term frequency inverse document frequency weighting).
b. Basic deep learning models: [30] Convolutional Neural Network **(CNN) [30]**, Long Short-Term Memory **(LSTM)** [31] and, a bidirectional LSTM model (**Bi-LSTM**) [31].
c. Models based on word embedding: **PV-DBOW** [32] (a paragraph vector model, the orders of words in text are ignored), **PV-DM** [30] (a paragraph vector model

proposed, which considers the word order), **fastText** [33] (an efficient model utilizing linear classifiers), **SWEM** [34] (introduced Word Embedding Model with Pooling Strategy), and **LEAM** [35] (Word Embedding Model with Attention Mechanism).

d. Graph-based deep learning models: **GCN-C** [36] (GCN model using Chebyshev filter), GCN-S [20] (GCN model using Spline filter), **GCN-F** [37] ((GCN model using Fourier filter) GCN model).

4.3. Compare Models

There are two comparison models:

1. **Text GCN** [7]: A short text classification method based on heterogeneous graphs between words and texts proposed by Yao et al.
2. **SHINE** [9]: Wang et al. proposed a short text classification method for words, POS tags, and heterogeneous graphs composed of entities extracted from external knowledge bases.

4.4. Comparative Test

The results in Table 2 show that the standard deviation of the deep learning-based models in group b is large, and the classification results are not stable. This is because these models are proposed for long text mining and cannot perform well in the face of short texts. The performance of the word embedding-based models in group c is different, which is consistent with the results of the above analysis. The word embedding-based model is limited in application and may not perform so well in all data. While the GNN model in group d is generally better than the other models in groups a, b, and c, which indicates that building graph models on text is effective. In the remaining four datasets, except the MR dataset, while the performance of the GNN model is the best, and GNN can accurately receive the information contained in the text and learn. Overall, PV-DM model and LSTM model perform poorly in prediction accuracy, and LSTM, pre-trained LSTM, and Bi-SLTM predict results with poor stability and high standard deviation. The TWPGCN model proposed in this paper is generally better than the existing models in terms of accuracy and standard deviation. The accuracy rates are improved by 1.3%, 0.87%, 0.65%, 1.94%, and 0.8% in the five datasets respectively. On the R52 dataset the largest increase. The standard deviation is also more or less improved, only slightly higher in the R52 dataset, and the predicted results in other datasets are relatively stable. The reason is that, since the R52 dataset is used for text classification of 52 categories for short texts, it is normal for the results to sway slightly.

Table 2. Test accuracy on document classification task. We run all models 10 times and report mean ± standard deviation.

Group	Model	20NG	R8	R52	Ohsumed	MR
a	TF-IDF + LR	0.8319 ± 0.0000	0.9374 ± 0.0000	0.8695 ± 0.0000	0.5466 ± 0.0000	0.7459 ± 0.0000
b	CNN-rand	0.7693 ± 0.0061	0.9402 ± 0.0057	0.8537 ± 0.0047	0.4387 ± 0.0100	0.7498 ± 0.0070
	CNN-non-static	0.8215 ± 0.0052	0.9571 ± 0.0052	0.8759 ± 0.0048	0.5844 ± 0.0106	0.7555 ± 0.0072
	LSTM	0.6571 ± 0.0152	0.9368 ± 0.0082	0.8554 ± 0.0113	0.4113 ± 0.0117	0.7506 ± 0.0044
	LSTM (pretrain)	0.7543 ± 0.0172	0.9609 ± 0.0019	0.9048 ± 0.0086	0.5110 ± 0.0150	0.7733 ± 0.0089
	Bi-LSTM	0.7318 ± 0.0185	0.9631 ± 0.0033	0.9054 ± 0.0091	0.4927 ± 0.0107	0.7768 ± 0.0086
c	PV-DBOW	0.7436 ± 0.0018	0.8587 ± 0.0010	0.7829 ± 0.0011	0.4665 ± 0.0019	0.6109 ± 0.0010
	PC-DM	0.5114 ± 0.0022	0.5207 ± 0.0004	0.4492 ± 0.0005	0.2950 ± 0.0007	0.5947 ± 0.0038
	PTE	0.7674 ± 0.0029	0.9669 ± 0.0013	0.9071 ± 0.0014	0.5358 ± 0.0029	0.7023 ± 0.0036
	fastText	0.7938 ± 0.0030	0.9613 ± 0.0021	0.9281 ± 0.0009	0.5770 ± 0.0049	0.7514 ± 0.0020
	fastText (bigrams)	0.7967 ± 0.0029	0.9474 ± 0.0011	0.9099 ± 0.0005	0.5569 ± 0.0039	0.7624 ± 0.0012
	SWEM	0.8516 ± 0.0029	0.9532 ± 0.0026	0.9294 ± 0.0024	0.6312 ± 0.0055	0.7665 ± 0.0063
	LEAM	0.8191 ± 0.0024	0.9331 ± 0.0024	0.9184 ± 0.0023	0.5858 ± 0.0079	0.7695 ± 0.0045

Table 2. *Cont.*

Group	Model	20NG	R8	R52	Ohsumed	MR
d	GCN-C	0.8142 ± 0.0032	0.9699 ± 0.0012	0.9275 ± 0.0022	0.6386 ± 0.0053	0.7722 ± 0.0027
	GCN-S	-	0.9680 ± 0.0020	0.9274 ± 0.0024	0.6282 ± 0.0037	0.7699 ± 0.0014
	GCN-F	-	0.9689 ± 0.0006	0.9320 ± 0.0004	0.6304 ± 0.0077	0.7674 ± 0.0021
	Text GCN	0.8634 ± 0.0009	0.9707 ± 0.0010	0.9356 ± 0.0018	0.6836 ± 0.0056	0.7674 ± 0.0020
	SHINE	-	-	-	0.4557	0.6458
	TWPGCN	0.8764 ± 0.0012	0.9794 ± 0.0007	0.9421 ± 0.0032	0.7030 ± 0.0021	0.7752 ± 0.0018

4.5. Ablation Experiment

In this paper, ablation experiments of small images and various heterogeneous images are carried out for the three proposed composition methods. As shown in Table 3 and Figure 4, the experiments are divided into three categories:

a. Take the W-W, W-P, T-W maps as the composition, input the GCN model, and predict the text.
b. Combining three small graphs in pairs to form a heterogeneous graph as input for prediction.
c. The TWPGCN model proposed in this paper.

Table 3. Ablation experiment.

Group	Model	20NG	R8	R52	Ohsumed	MR
a	W-W	0.8671	0.9772	0.9350	0.6760	0.7721
	W-P	0.8573	0.9703	0.9380	0.6777	0.7684
	T-W	0.8455	0.9584	0.9272	0.6688	0.7665
b	W-W+T-W	0.8657	0.9689	0.9385	0.6820	0.7696
	W-W+W-P	0.8655	0.9767	**0.9428**	0.6962	0.7743
	W-P+T-W	0.8728	0.9762	0.9326	0.6860	0.7676
c	TWPGCN	**0.8764**	**0.9794**	0.9421	**0.7030**	**0.7752**

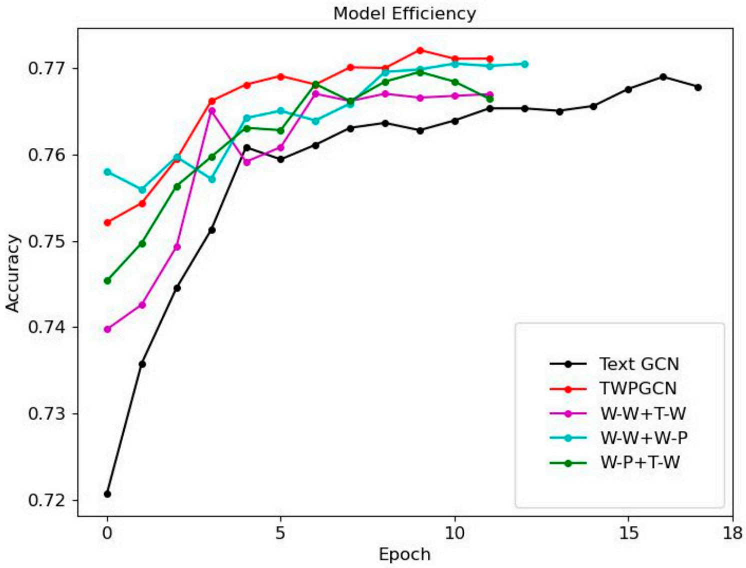


Figure 4. Efficiency comparison experiment.

Through the ablation experiment, it can be found that the result of the heterogeneous graph combined is better than that of the small graph in general, because the addition of more nodes and edges improves the information content of the graph. From the experimental results of group a, it seems that inputting W-W graph for the model has better results. It represents the co-occurrence relationship between words in the text, which shows that the recognition of fixed collocations and special words is very important in short text classification. In group b, the experimental effect of the heterogeneous graph composed of W-W and W-P is better. The addition of POS tags provides the attributes of different words for the word co-occurrence graph, which enables the model to have a better ability to distinguish important nodes. TWPGCN integrates both word co-occurrence and label features into text features, and the enhanced word features transmit information to text nodes, which makes the information contained in the text nodes more recognizable and classification clearer. Experimental data show that TWPGCN is the most expressive for short text classification in the four datasets, except R52.

4.6. Model Efficiency

In this paper, a model efficiency comparison experiment is performed on the MR dataset for the Text GCN model that performs the best in the comparison test. The experimental results are shown in Table 4 and Figure 4.

Table 4. Efficiency comparison experiment.

Model	Convergent Epoch	Avg. Epoch Time /s	Total Time /s
Text GCN	17	1.03	17.53
W-W + T-W	12	1.05	12.62
W-P + T-W	11	0.58	6.48
W-W + W-P	12	1.06	12.71
TWPGCN	12	1.09	13.04

From Table 4, it can be found that the W-P + T-W model takes the shortest time, only 6.48 s, because the W-P + T-W model only builds edges between words and documents, as well as documents and labels, and the constructed graph smaller, the number of input features is smaller, so the model training speed is fast. But, just the training speed is not enough, it can be seen from Figure 4 that the W-P + T-W model is relatively unstable and the model accuracy is not high. It should be trained together with text features in short text classification. The average time of each epoch of the Text GCN model is shorter than that of TWPGCN, which only takes 1.03 s, but it takes 17.53 s to reach the convergence state at the 17th epoch, while the TWPGCN has converged at the 12th epoch, which takes 13.04 s, which is 4.49 s faster than Text GCN. This shows that TWPGCN has an increase in model complexity, and each training takes a long time, but in general, the model training efficiency is improved, which improves the calculation speed.

As can be seen from Figure 4, in the first 4 epochs, the accuracy of all models improved rapidly, and the models quickly learned and adapted to the input features in the early stage. After the 8th epoch, the accuracy of the model improved slowly, and reached the convergence ends before and after the highest accuracy. In general, the Text GCN model requires the longest training period and has the lowest accuracy. Among the models proposed in this paper, the W-W + W-P model requires a longer training period, and the W-W + T-W model is relatively unstable, the TWPGCN model performs the best. The Text GCN model progresses slowly in the middle of training, and the improvement is not obvious. After the 17th epoch reaches the optimal accuracy, the model converges at the 18th epoch. In contrast, the TWPGCN model did not reduce its progress in the middle of training. It reached the best and most accurate rate in the 9th epoch and converged in the 11th epoch. The accuracy rate at the time of convergence is still the best. In summary, the TWPGCN model proposed in this paper has the advantages of high efficiency, high accuracy, and high precision.

5. Discussion

In today's explosive growth of information, the types of text are also divided into more and more fine-grained, in addition to short texts, there are ultra-short texts and ultra-long texts. This paper focuses on short text classification, mainly because short texts are in the majority on the web, and commentary and news texts are included in the scope of this paper. Subsequent research will be more detailed, and the analysis of comments with emojis or the identification of many fake news on the Internet is a more challenging topic. In the professional field, the classification of paper abstracts is also an interesting application. The paper abstracts contain professional vocabulary and specific phrases, which will require greater word embedding, and will also be a test for the learning and discrimination ability of the model. As the basis of these studies, short text analysis should achieve higher precision, and analyzing short texts from various angles can provide some ideas for the following research.

After a long period of study and research, the accuracy of traditional models is not low in the experiments, but it is precisely because has the research gradually matured that it seems to have reached a bottleneck period, and emerging deep learning models are increasingly replacing traditional models. Deep learning enables models to learn and adapt to a wide variety of tasks. Due to different learning methods, the models are good at different fields. CNN models that use adjacent features are good at images, while RNN models that focus on temporal features are more used for text. Thanks to the compositional flexibility of GNN, targeted GNN models can solve more specific tasks, short text classification is one of them. Word embedding technology is an important milestone for the development of NLP, and there are many excellent papers around it. Similar to traditional models, its limitations are gradually exposed as time progresses. This article chooses not to use pre-trained word vectors because of these considerations. Perhaps future research can make more breakthroughs in word embedding, and then combine word embedding with the TWPGCN proposed in this article or other existing models, there will be more better results.

In the introduction of this paper, we analyze the research status of short text classification. According to the current research, there is a lack of text classification methods suitable for short texts. Traditional text classification methods are not targeted enough, and methods based on pre-training and external knowledge bases are not necessarily suitable for all occasions. Generally speaking, methods based on GNN are more suitable for short text classification, the TWPGCN proposed in this paper is based on GNN.

First of all, this paper composes graph for short texts, and extracts text, words, and POS tags in short texts as nodes to construct three kinds of graphs, namely W-W graph, W-P graph, and T-W graph. Then, based on these small graphs, a heterogeneous graph suitable for short text classification is constructed, and a TWPGCN suitable for the above heterogeneous graph is designed to learn it. After that, this paper conducts a comparative experiment on the existing short text classification methods, and analyzes the advantages and disadvantages of various models. In this paper, an ablation experiment is carried out for the proposed method, and the data is experimentally compared on the W-W graph, W-P graph and T-W graph, and the W-W + T-W graph, W-W + W-P after combining the three small graphs in pairs. Graph W-P + T-W are also experimented, and the ablation experiments analyze the advantages and disadvantages of small graph and heterogeneous graph, and compare with the TWPGCN proposed in this paper. The text also conducts model efficiency experiments, analyzes the experimental time and efficiency of various graphs, and compares it with the existing excellent short text classification model Text GCN.

The model proposed in this paper focuses on short texts, and selects important text features, word features, and POS tag features. Compared with traditional models, it is more suitable for short texts and has achieved better results in experiments. Compared with traditional models, it is more effective. However, the single-layer GNN is difficult to learn and takes a long time, which has to be admitted as a deficiency. How to improve the efficiency of a single-layer network without increasing the difficulty of learning on a targeted basis is also a research direction to be considered in the future.

The TWPGCN proposed in this paper cannot only be used for short text classification, but also provides a way of thinking for GNN. The combination of small graphs and large graphs can focus on important nodes of network learning in the way of composition. The reason why short text is selected as the application of this model is that the information obtained in short text analysis is less, and the key points are more difficult to capture, and the information of text is mainly reflected in the entities and the features they contain. Experiments have proved that although the GNN takes a long time in a single learning, the overall efficiency has been improved, and the accuracy has also increased, once again affirming the position of the GNN in deep learning. Future research will use GNN for more and more complex applications to solve more difficult problems in a targeted manner.

6. Conclusions

Aiming at the characteristics of short text samples such as sparse features, lack of context, and too much noise data, this paper proposes TWPGCN to classify short texts, and has achieved good experimental results in five benchmark datasets. Pros and cons in classification. TWPGCN does not require pre-trained language models, word embedding, and external knowledge bases, which reduces the computational complexity and speed of short text classification and makes the model pay more attention to the text itself. If other scholars want to use pre-trained language models, word vector embedding, and external knowledge bases, they should consider how to reduce the noise impact of external knowledge on texts, and train pre-trained models with a wider range of applications.

This paper does not use pre-trained language models, word embeddings, and external knowledge bases because of the noise and other effects they bring. If future researchers hope to make breakthroughs in this area, they should consider how to reduce the adverse effects of external knowledge on the text, improve the ability of the model to select key features, or train a pre-training model with a wider range of applications. For further researchers, you can also refer to this article for the short text itself, how to focus on more important information and learn from short texts. In the future, the classification of short texts will be more refined, focusing on special fields, such as classification of paper abstracts, news texts, ultra-short texts or social network texts. Short texts in special fields will face more complex challenges, requiring more targeted models and more efficient text processing methods. The design of the GNN should also be more flexible, and a more general or more targeted GNN will be a difficulty in future research.

Author Contributions: Conceptualization, B.Z.; methodology, B.Z.; software, B.Z.; validation, B.Z., Q.H. and D.Z.; formal analysis, Q.H. and D.Z.; investigation, B.Z.; resources, Q.H. and D.Z.; writing—original draft preparation, B.Z.; writing—review and editing, B.Z. and Q.H.; visualization, B.Z. All authors have read and agreed to the published version of the manuscript.

Funding: The research was funded by the National Natural Science Foundation of China "Research on the Evidence Chain Construction from the Analysis of the Investigation Documents (62166006)", supported by Guizhou Provincial Science and Technology Projects (Guizhou Science Foundation-ZK [2021] General 335), supported by Guizhou Provincial Science and Technology Projects (Guizhou Province under Grant [2020]1Y254).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The study did not report any data.

Conflicts of Interest: There are no conflict to declare.

References

1. Li, J.; Zhang, D.; Wulamu, A. Investigating Multi-Level Semantic Extraction with Squash Capsules for Short Text Classification. *Entropy* **2022**, *24*, 590. [CrossRef] [PubMed]
2. Yang, K.; Miao, R. Research on Improvement of Text Processing and Clustering Algorithms in Public Opinion Early Warning System. In Proceedings of the 2018 5th International Conference on Systems and Informatics (ICSAI), Nanjing, China, 10–12 November 2018; pp. 333–337. [CrossRef]

3. Wang, J.; Wang, Z.; Zhang, D.; Yan, J. Combining Knowledge with Deep Convolutional Neural Networks for Short Text Classification. In Proceedings of the International Joint Conference on Artificial Intelligence, Melbourne, Australia, 19–25 August 2017; Volume 350, pp. 3172077–3172295.
4. Maji, S.; Berg, A.C.; Malik, J. Efficient classification for additive kernel SVMs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *35*, 66–77. [CrossRef] [PubMed]
5. Zhang, M.-L.; Peña, J.M.; Robles, V. Feature selection for multi-label naive Bayes classification. *Inf. Sci.* **2009**, *179*, 3218–3229. [CrossRef]
6. Bijalwan, V.; Kumar, V.; Kumari, P.; Pascual, J. KNN based Machine Learning Approach for Text and Document Mining. *Int. J. Database Theory Appl.* **2014**, *7*, 61–70. [CrossRef]
7. Yao, L.; Mao, C.; Luo, Y. Graph convolutional networks for text classification. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 7370–7377.
8. Tayal, K.; Rao, N.; Agarwal, S.; Jia, X.; Subbian, K.; Kumar, V. Regularized graph convolutional networks for short text classification. In Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, Spain, 8–13 December 2020; pp. 236–242.
9. Wang, Y.; Wang, S.; Yao, Q.; Dou, D. Hierarchical Heterogeneous Graph Representation Learning for Short Text Classification. *arXiv* **2021**, arXiv:2111.00180. [CrossRef]
10. Zhao, H.; Xie, J.; Wang, H. Graph Convolutional Network Based on Multi-Head Pooling for Short Text Classification. *IEEE Access* **2022**, *10*, 11947–11956. [CrossRef]
11. Cover, T.; Hart, P. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory* **1967**, *13*, 21–27. [CrossRef]
12. Maher, P.E.; Clair, D.S. Uncertain Reasoning in an ID3 Machine Learning Framework. In Proceedings of the Second IEEE International Conference on Fuzzy Systems, San Francisco, CA, USA, 28 March–1 April 1993; pp. 7–12.
13. Loh, W.Y. Classification and regression tree methods. *Encycl. Stat. Qual. Reliab.* **2008**, *1*, 315–323.
14. Salzberg, S.L. C4. 5: Programs for machine learning by j. ross quinlan. *Morgan Kaufmann Publ.* **1993**, *1*, 235–240.
15. Wang, P.; Xu, B.; Xu, J.; Tian, G.; Liu, C.-L.; Hao, H. Semantic expansion using word embedding clustering and convolutional neural network for improving short text classification. *Neurocomputing* **2016**, *174*, 806–814. [CrossRef]
16. Shi, K.; Li, L.; Liu, H.; He, J.; Zhang, N.; Song, W. An Improved KNN Text Classification Algorithm Based on Density. In Proceedings of the 2011 IEEE International Conference on Cloud Computing and Intelligence Systems, Beijing, China, 15–17 September 2011; pp. 113–117.
17. Zhang, H.; Zhong, G. Improving short text classification by learning vector representations of both words and hidden topics. *Knowl.-Based Syst.* **2016**, *102*, 76–86. [CrossRef]
18. Kettaf, C.; Yousfate, A. Authorship Attribution by Functional Discriminant Analysis. In *International Conference on Mathematical Aspects of Computer and Information Sciences*; Springer: Cham, Switzerland, 2019; pp. 438–449.
19. de Arruda, H.F.; Marinho, V.Q.; Costa, L.F.; Amancio, D.R. Paragraph-based complex networks: Application to document classification and authenticity verification. *arXiv* **2018**, arXiv:1806.08467.
20. Bruna, J.; Zaremba, W.; Szlam, A.; LeCun, Y. Spectral networks and locally connected networks on graphs. *arXiv* **2013**, arXiv:1312.6203.
21. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
22. Liang, J.; He, Q.; Zhang, D.; Fan, S. Extraction of Joint Entity and Relationships with Soft Pruning and Global Pointer. *Appl. Sci.* **2022**, *12*, 6361. [CrossRef]
23. Li, H.; Li, W.; Zhang, H.; He, X.; Zheng, M.; Song, H. Automatic Image Annotation by Sequentially Learning from Multi-Level Semantic Neighborhoods. *IEEE Access* **2021**, *9*, 135742–135754. [CrossRef]
24. Korban, M.; Li, X. DDGCN: A Dynamic Directed Graph Convolutional Network for Action Recognition. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 761–776. [CrossRef]
25. Chen, G.; Ye, D.; Xing, Z.; Chen, J.; Cambria, E. Ensemble application of convolutional and recurrent neural networks for multi-label text categorization. In Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 14–19 May 2017; pp. 2377–2383. [CrossRef]
26. Lang, K. NewsWeeder: Learning to Filter Netnews. In *Machine Learning Proceedings 1995*; Morgan Kaufmann: Burlington, MA, USA, 1995; pp. 331–339. [CrossRef]
27. Hersh, W.; Buckley, C.; Leone, T.J.; Hickam, D. OHSUMED: An Interactive Retrieval Evaluation and New Large Test Collection for Research. In *SIGIR '94*; Springer: London, UK, 1994; pp. 192–201. [CrossRef]
28. Linmei, H.; Yang, T.; Shi, C.; Ji, H.; Li, X. 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), Hong Kong, China, 3–7 November 2019; pp. 4821–4830.
29. Pang, B.; Lee, L. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In Proceedings of the Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05), Ann Arbor, MI, USA, 25–30 June 2005.
30. Chen, Y. Convolutional Neural Network for Sentence Classification. Master's Thesis, University of Waterloo, Waterloo, ON, Canada, 2015.
31. Liu, P.; Qiu, X.; Huang, X. Recurrent neural network for text classification with multi-task learning. *arXiv* **2016**, arXiv:1605.05101, 2016.
32. Le, Q.; Mikolov, T. Distributed representations of sentences and documents. In Proceedings of the International Conference on Machine Learning, Beijing, China, 21–26 June 2014; pp. 1188–1196.
33. Joulin, A.; Grave, E.; Bojanowski, P.; Mikolov, T. Bag of tricks for efficient text classification. *arXiv* **2016**, arXiv:1607.01759.

34. Shen, D.; Wang, G.; Wang, W.; Min, M.R.; Su, Q.; Zhang, Y.; Li, C.; Henao, R.; Carin, L. Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms. *arXiv* **2018**, arXiv:1805.09843. [CrossRef]
35. Wang, G.; Li, C.; Wang, W.; Zhang, Y.; Shen, D.; Zhang, X.; Henao, R.; Carin, L. Joint Embedding of Words and Labels for Text Classification. *arXiv* **2018**, arXiv:1805.09843. [CrossRef]
36. Defferrard, M.; Bresson, X.; Vandergheynst, P. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2016; Volume 29.
37. Henaff, M.; Bruna, J.; LeCun, Y. Deep convolutional networks on graph-structured data. *arXiv* **2015**, arXiv:1506.05163.