applied sciences

MDPI

Article

Heterogeneous Graph-Convolution-Network-Based Short-Text Classification

Jiwei Hua *, Debing Sun, Yanxiang Hu, Jiayu Wang, Shuquan Feng and Zhaoyang Wang

The School of Computer and Information Engineering, Tianjin Normal University, Tianjin 300387, China; sundebing@tjnu.edu.cn (D.S.); huyanxiang@tjnu.edu.cn (Y.H.); wangjiayu908@163.com (J.W.); 2211090042@stu.tjnu.edu.cn (S.F.); wangzhaoyang@stu.tjnu.edu.cn (Z.W.)

* Correspondence: huajiwei@tjnu.edu.cn; Tel.: +86-138-0211-3383

Abstract: With the development of online interactive media platforms, a large amount of short text has appeared on the internet. Determining how to classify these short texts efficiently and accurately is of great significance. Graph neural networks can capture information dependencies in the entire short-text corpus, thereby enhancing feature expression and improving classification accuracy. However, existing works have overlooked the role of entities in these short texts. In this paper, we propose a heterogeneous graph-convolution-network-based short-text classification (SHGCN) method that integrates heterogeneous graph convolutional neural networks of text, entities, and words. Firstly, the model constructs a graph network of the text and extracts entity nodes and word nodes. Secondly, the relationship of the graph nodes in the heterogeneous graphs is determined by the mutual information between the words, the relationship between the documents and words, and the confidence between the words and entities. Then, the feature is represented through a word graph and combined with its BERT embedding, and the word feature is strengthened through BiLstm. Finally, the enhanced word features are combined with the document graph representation features to predict the document categories. To verify the performance of the model, experiments were conducted on the public datasets AGNews, R52, and MR. The classification accuracy of SHGCN reached 88.38%, 93.87%, and 82.87%, respectively, which is superior to that of some existing advanced classification methods.

Keywords: short-text classification; physical information; graph convolution neural network; BERT

check for updates

Citation: Hua, J.; Sun, D.; Hu, Y.; Wang, J.; Feng, S.; Wang, Z. Heterogeneous Graph-Convolution-Network-Based Short-Text Classification. *Appl. Sci.* **2024**, *14*, 2279. https://doi.org/10.3390/app14062279

Academic Editor: Andrea Prati

Received: 14 November 2023
Revised: 26 January 2024
Accepted: 2 February 2024
Published: 8 March 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Due to the rapid development of the internet, people receive a large number of short-text messages such as instant messages, news reports, film and television reviews, and business exchanges through various applications and web pages. It is particularly important to effectively process these pieces of content and mine useful information from them. Due to the lack of necessary logic between short-text sentences, short-text classification, which is a basic task in natural language processing (NLP) [1], plays an important role in text information processing in fields such as dialogue and question answering, emotion analysis [2], and public opinion analysis. Unlike long text, short-text data have the following characteristics:

(1) Semantic sparsity: Short text contains fewer words and fewer words with actual semantics compared with long text, which makes it difficult to extract useful information for classification.
(2) Sentence irregularity: Most short-text sentences, such as news headlines, conversation messages, and microblogs, are close to daily life. They have the characteristics of concise expression, colloquial sentence style, extensive use of network buzzwords, etc., which pose a great challenge for the accurate recognition of classifiers.
(3) Large data scale: Massive short-text data have flooded the network, so traditional manual data processing methods can no longer meet the real-time data processing requirements.

The above characteristics of short text make it difficult to apply traditional text classification methods to short-text data. Recently, deep learning models such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have been applied to short-text classification. Compared with traditional methods such as logical regression [3] and support vector machines (SVMs) [4], deep learning models can give priority to the local and sequential characteristics of the text and show good results in terms of classification. However, these above-mentioned deep learning models often ignore the global features of the corpus when modeling short text.

In recent years, graph neural networks (GNNs) [5] have attracted extensive attention from researchers because they can effectively deal with text structures with complex relationships and retain global word features. For example, the TextGCN model learns the feature representation of text for classification tasks by constructing a heterogeneous graph of the text and words. However, due to the small number of short-text words and the lack of contextual semantic information, TextGCN has a poor classification effect on short-text datasets.

In order to solve the problem of the sparse semantic features of short text, this paper proposes a heterogeneous graph-convolution-network-based short-text classification (SHGCN) method, which integrates external entity information, text, and words into the graph neural network to model short text. It not only captures the relationship between text and entities, but also learns the feature representation of text and words. Further, in this model, the learned text representation and word representation are input into Bi-LSTM; as a result of the combination of the BERT word embedding representation, the short text is perfectly classified by the Bi-LSTM model. The experimental results show that SHGCN has a higher detection accuracy compared to other commonly used short-text classification methods.

The main innovations of this paper are as follows:

(1) This paper proposes a method of integrating entities into graph-based network modeling of short texts, which can eliminate the ambiguity of some words in the text.
(2) A heterogeneous graph-convolution-network-based short-text classification (SHGCN) method that integrates external entities is proposed; it can utilize external entity information to mine the potential semantics of text and more accurately learn the representation of text and word features.
(3) The model was validated on three public datasets, i.e., AGNews, R52, and MR, and the experimental results showed that the classification performance of the model was superior to that of other mainstream baseline methods.

This paper first introduces the research background; Section 2 introduces the related works of text classification; Section 3 introduces the entity extraction module, embedded input module, feature learning module, and category output module of the model; Section 4 introduces the experimental setup and analyzes the results; Section 5 summarizes the full text.

2. Related Work

Short-text classification aims to select appropriate labels for a large number of unmarked texts. The existing text classification methods can be divided into three categories: statistic-based methods, deep learning methods, and graph-neural-network-based methods.

A statistical text classification algorithm needs the design and classification algorithm of feature engineering. Feature engineering processes the text data, extracts them as features, and uses them as the input of subsequent classifiers. Usually, the word bag model [6] is used to obtain the data features. In addition, there are some complex text feature projects. For example, the n-gram model proposed by Wang et al. [7] is based on the algorithm of statistical language, which divides the text into a byte fragment sequence (also named a gram) according to the sliding window with the length of n-grams that occur more frequently from a list, which serves as the feature vector space of the text. The topic model proposed by Wallach et al. [8] combines the n-gram model with potential topic variables,

forming a hierarchical Dirchlet binary generation model that can more accurately generate potential topics that are not affected by function words and obtain a deep representation of the text. Classification algorithms generally include logical regression (LR), support vector machine (SVM), gradient lifting decision tree (GLDT) [9], etc. But text feature engineering often relies on human processing data. For massive data, the cost is too high, the processing time is too long, the text representation obtained by traditional methods has the characteristics of high dimensionality and high sparsity, and the feature expression ability is weak, which is not conducive to classification tasks.

Deep learning has made a breakthrough in the field of short-text classification. Compared with traditional text classification algorithms, text classification algorithms based on deep learning can learn deeper and more complex features of text and can achieve end-to-end processing by automatically extracting text features, eliminating complicated manual operations. TextCNN, proposed by Kim et al. [5], applied a convolutional neural network to text classification tasks to capture local information between texts through multiple convolutional cores. The TextRNN proposed by [10] could capture the context semantic relationship of the sequence as long as possible. However, in the training process of the RNN, the gradient would disappear, and thus, it is hard for the long-distance sequence information to be learned. To solve the problem of sparse short-text data, Zeng et al. [11] proposed a topic memory network that did not rely on external corpora; it could find keywords for classification through the word co-occurrence feature of the entire dataset and mine potential topics for classification. Li et al. [12] proposed a model of a convolutional neural network based on knowledge-powered attention with a similarity matrix (KASM) that used a knowledge map (KG) to enrich the semantic representation of short text. These methods only modeled the local information of the text, without paying attention to the global information of the text.

In recent years, graph neural networks [13] have been applied to text classification tasks. Yao et al. [14] proposed a text graph convolutional network (Text GCN) that establishes a heterogeneous text graph for the entire corpus through word co-occurrence information and document word relationships, after which the representations of the document and word nodes are learned. Tensor GCN [15] was proposed on the basis of TextGCN. Heterogeneous graphs were built based on semantics, syntax, and sequence, and node information could be spread between graphs. Yang T et al. [16] proposed a heterogeneous graph attention network classification model that takes text, topics, and entities as nodes and constructs edges between text nodes and topic nodes, text nodes and entity nodes, and entity nodes and entity nodes. Reference [17] combines mutually exclusive sentence-level co-occurrences to form a document-level graph and uses structure learning to sparsely select edges with dynamic context dependency. Reference [18] proposes that in the process of constructing a conductive text classification model through graph neural networks, attention mechanisms are used to fuse the structural semantics in heterogeneous graphs. Reference [19] used graph neural networks for label propagation and inference to achieve semi-supervised short-text classification tasks. The above GCN-based works can lead to information redundancy and a lack of context awareness. Yang S.G. et al. [20] proposed a graph attention network that integrated node and edge weight values. The gravity model (GM) was used to evaluate the importance of word nodes, and the weight of the edge was obtained through point mutual information (PMI), which was then applied to text classification. However, the above methods only use corpus information to build text graphs, which failed to solve the problem of the sparse features of short text.

3. Model Description

In order to solve the problem of sparse semantic space of short texts and learn the feature representation of documents and the words in documents, this paper proposes heterogeneous graph-convolution-network-based short-text classification (SHGCN). As shown in Figure 1, the structure of SHGCN includes four modules: the entity link module, the embedded input module, the feature learning module, and the category output module.

The entity link module maps the words in the short text to the entities in Wikipedia through the entity link tool. The document embedding, entity embedding, and word embedding of the embedded input module map the documents, the entities, and the words to the high-dimensional vector space, respectively.

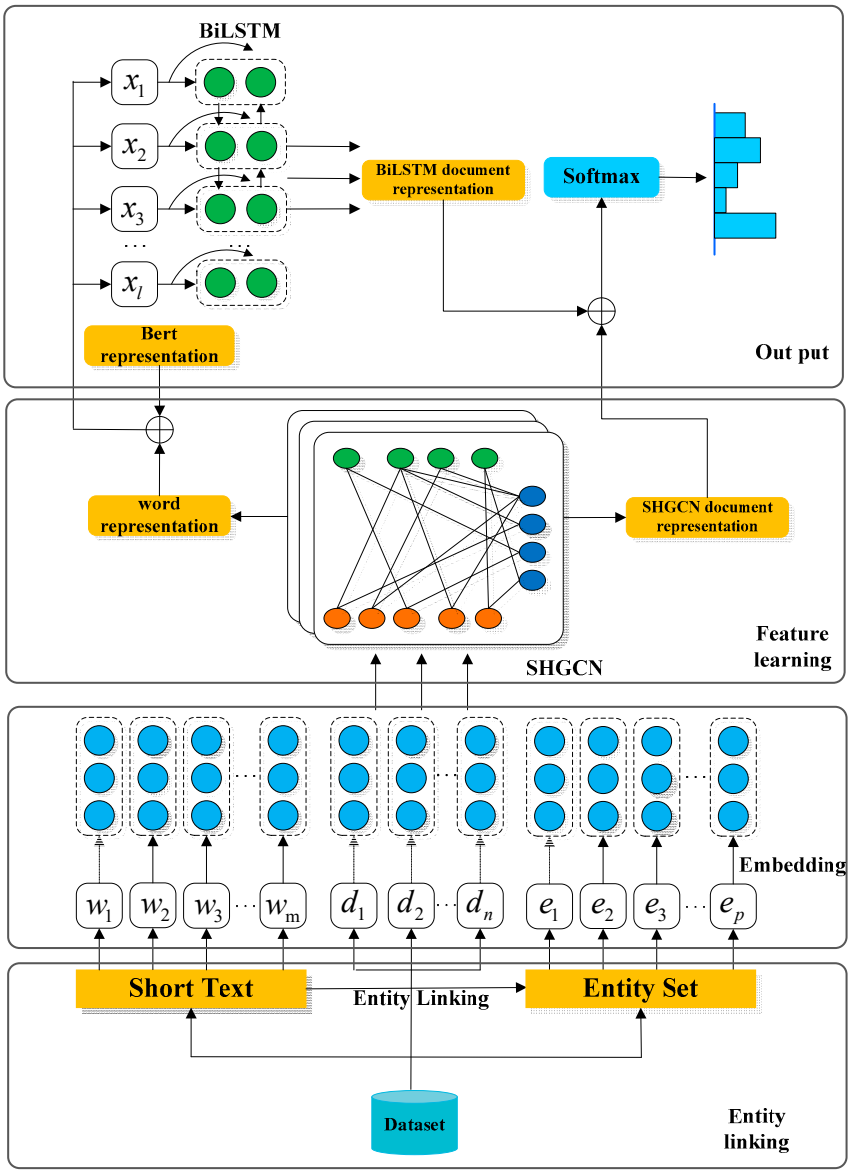

Figure 1. Structure of heterogeneous graph-convolution-network-based short-text classification (SHGCN).

The feature learning module uses the heterogeneous graph convolution neural network to train the input embedded features, with the aim of learning the document feature representation and the word feature representation. The category output module fuses the word feature representation learned by the feature learning module with the BERT pre-training word embedding feature. The fused features serve as the input of BiLSTM. BiLSTM is used to capture the features of the text context. Finally, the obtained hidden-state features are spliced with the document features obtained by the feature learning module, and then the category of the short text is obtained by linear transformation.

3.1. The Entity Link Module

Entity linking can solve the problem of conceptual ambiguity and annotation of short-text vocabulary, so as to further enrich the expression of short text. TagMe is one of the best

entity-linking tools in the scientific community, especially in the annotation of short texts. Unlike word embedding, this paper uses the TagMe entity-linking tool [21] to map words to entities in Wikipedia and uses an external knowledge base to expand the word concept of short text.

3.2. The Embedded Input Module

Since word embedding can capture lexical semantics in digital form and process abstract semantic concepts, it has been widely used in text classification, question-answering systems, knowledge mining, and other fields [22]. Word2Vec [23,24] and Glove [25] are two commonly used word embedding methods in text classification tasks. Through a sliding window containing local context information, Word2Vec can capture the semantics of words, mine the correlation between words, and obtain the reprehensive features of words. Glove can capture the global semantic information of words based on the global word co-occurrence matrix. In this study, words, entities, and documents are mapped to a high-dimensional vector to form the features of the graph neural network nodes. For word nodes, randomly initialized features are used as their features. The pre-trained Wikipedia entity features are taken as features of the entity nodes. For the document nodes, the average of the pre-trained word embedding values of all the words in the document is used as their feature.

3.3. The Feature Learning Module

In most existing studies, researchers only obtain information from the corpus. However, the characteristics of the short text make it challenging to obtain sufficient semantic information from short texts. This paper fully considers the external entity information and establishes a heterogeneous graph $G = (v, \varepsilon)$, in which v is the node set and ε is the edge set. $v = D \cup E \cup W$ denotes that *it* consists of three parts: the document node set $D = \{d_1, d_2, \ldots, d_n\}$, the entity node set $E = \{e_1, e_2, \ldots, e_n\}$, and the word node set $W = \{w_1, w_2, \ldots, w_n\}$. ε indicates the relationship between nodes.

The edge between a document node and a word node is determined by the frequency of the word in the document. That is, term frequency–inverse document frequency (TF-IDF) is used as the weight of the edge between a document node and a word node. The value of the edge between two word nodes is determined by the word co-occurrence information of the whole corpus, which uses the point mutual information (PMI) between words [26] to evaluate the degree of correlation between words as the weight of the edge between the two word nodes. The value of the edge between the document nodes and the entity nodes is determined by the confidence of the words in the document mapped to Wikipedia entities. For node i and node j in G, the adjacency matrix A_{ij} is defined as follows:

$$A_{ij} = \begin{cases} TF - IDF_{ij} & i \text{ is document}, j \text{ is word} \\ PMI(i,j) & i, j \text{ are words} \\ Score_{ij} & i \text{ is document}, \ j \text{ is entity} \\ 1 & i = j \\ 0 & otherwise \end{cases} \tag{1}$$

Using the sliding window, the $PMI(i,j)$ value between word i and word j in each text of the corpus is calculated as follows:

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

where $p(i,j)$ is the proportion of the number of sliding windows in which words i and j appear simultaneously in the corpus to the total number of sliding windows. $p(i)$ is the proportion of the number of sliding windows with i in the corpus, and $p(j)$ is the proportion of the number of sliding windows with j in the corpus. The higher the PMI value, the greater

the semantic correlation between two words. The value of PMI is positively correlated with the semantic correlation of words in the corpus.

The feature matrix of the document–entity–word node is defined as $X \in R^{c \times d}$. For the one-layer GNN network, the features of the K-dimensional document–entity–word node can be expressed as follows:

$$L^{(1)} \in \rho(\widetilde{A}XW_0) \tag{3}$$

where $\widetilde{A} = D^{-1/2}AD^{1/2}$ is a regularized and normalized adjacency matrix, A is the adjacency matrix of the text heterogeneous graph, D is the degree matrix, W_0 is the weight matrix, and $\rho(\cdot)$ is the activation function. The multi-layer GNN aggregates the information of neighbor nodes of different orders, and the output characteristics can be expressed as follows:

$$L^{(j+1)} \in \rho(\widetilde{A}L^{(j)}W_j) \tag{4}$$

where j is the number of layers and $L^{(0)} = X$. In this study, we use the two-layer graph convolution neural network to learn the node features and obtain the feature representation of document, entity, and word nodes.

3.4. The Category Output Module

The TextGCN model does not perform as well as CNN and LSTM in the MR dataset. This is because it does not consider the word order in short-text classification. Therefore, to fully consider the characteristics of word order in short text and improve the accuracy of the model, the model proposed in this paper further classifies the representation of document nodes and word nodes learned by graph convolution neural network through the BiLSTM model, and it uses the final output as the final prediction of the short-text category.

Since the BERT pre-training model can effectively generate word embedding containing contextual semantic information, SHGCN proposes splicing the word node obtained by the feature learning module $Rw = \{w_1, w_2, ...w_l\}$ and the word node obtained by the BERT pre-training module $Rw' = \{w_1^{'}, w_2^{'}, ...w_l^{'}\}$ as the input H of the BiLSTM. We have

$$H = concat(Rw, Rw^{'}) \tag{5}$$

where $concat(\cdot)$ is the splicing process. The node hidden state h output by BiLSTM is spliced with the document node feature Rs learned by the GCN, and the result is then fed into a softmax classifier to obtain the text prediction label γ':

$$\gamma^{'} = softmax(concat(h, Rs)) \tag{6}$$

Finally, cross-entropy loss is used to train the final classification results as follows:

$$l = CrossEntropy(\gamma, \gamma^{'}) \tag{7}$$

where γ is the real label of short text.

4. Experiment and Performance Analysis

This section verifies the performance of the proposed model by comparing different short-text classification methods. The datasets, comparison models, and parameters are as follows:

4.1. Datasets

We conducted experiments on three widely used short-text datasets: AGNews, R52, and Movie Review (MR).

AGNews: AGNews contains English news that consists of four categories: World, Sports, Business, and Sci/Tec. Each category contains 30,000 training samples and 1900 test samples.

R52: R52 is a subset of Reuters 21,578 datasets, with 52 categories, 6532 training samples, and 2568 test samples.

MR: MR is a binary emotional dataset of film reviews; it contains 5331 positive and 5331 negative comments. There are 7108 training samples and 3554 testing samples.

For this paper, we used the NLTK library to remove the stop words and remove the words that appear fewer than five times in both the AGNews and R52 datasets. Since texts in the MR dataset are too short, no stop words and low-frequency words need to be removed. The datasets after preprocessing are shown in Table 1.

Table 1. Summary statistics of datasets.

Dataset	Training Set	Test Set	Category	Average Number of Entities	Average Length	Total Number of Words	Average Number of Words	Category Proportion
AGNews	120,000	7600	4	5.59	23.36	302,210	37.78	world: 0.25 sports: 0.25 business: 0.25 science: 0.25
R52	6532	2568	52	8.71	69.82	997,060	109.57	trade: 0.036 earn: 0.431 jobs: 0.005 ship: 0.016 . . .. . .
MR	7108	3554	2	1.15	20.39	224,073	21.02	positive: 0.5 negative: 0.5

4.2. Baselines

Six text classification baseline models were selected for comparison with the proposed ETGCN model proposed in this paper. The baseline models are described as follows:

TF-IDF+LR: The bag-of-words model with word frequency–inverse document frequency weighting, using logistic regression as a classifier.

CNN [9]: Convolutional neural network. In this study, Glove pre-trained word embedding is used as the input of the CNN.

BiLSTM [10]: A bidirectional LSTM model for text classification. In this study, Glove pre-trained words are embedded into the model.

fastText [21]: Facebook's open-source fast text classification tool; it selects the average value of all word embeddings as text embedding.

TextGCN [14]: TextGCN is a graph convolution neural network used for text classification. It uses randomly initialized word embedding as input.

TensorGCN [15]: A graph convolution neural network based on a graph tensor formed by semantics, syntax, and sequence text graphs.

4.3. Parameter Setting

The experiment used the PyTorch framework, and the data were trained and tested on an NVIDIA GTX 1080Ti GPU(NVIDIA, Santa Clara, CA, USA). The number of convolutional layers in the SHGCN model was set to two, the optimizer was Adam, and the loss function was cross-entropy. If the performance of a model did not decline for 10 consecutive epochs, the training was terminated in advance.

4.4. Experimental Results and Analysis

The effects of the sliding window size and word embedding dimension on the accuracy of the model were tested. With other parameters unchanged, the size of the sliding window was set as 5, 10, 15, 20, 25, or 30. The SHGCN model was used to conduct experiments on the test set of MR And AGNews datasets. The results are shown in Figure 2. With the increase in the size of the sliding window, the accuracy rate increases at first and then

decreases. This is because if the size of the sliding window is too small, the connections between long-distance words will be ignored, resulting in a failure to capture more co-occurrence information between words, and therefore, the performance of the model will be affected. On the contrary, if the size of the sliding window is too large, it will establish relations with some words with weak semantic relevance and generate unnecessary noise, which will eventually affect the classification results.

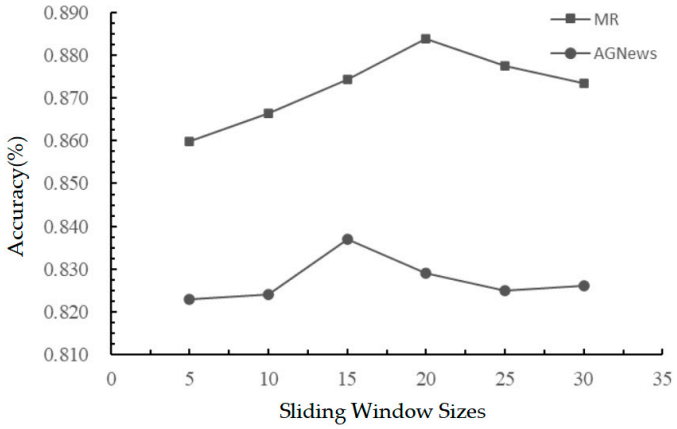

Figure 2. Accuracies of SHGCN with different sliding window sizes.

With other parameters unchanged, the embedding dimension was set as 50, 100, 150, 200, 250, or 300 respectively. As shown in Figure 3, when the word embedding dimension is 150, the accuracy rate of SHGCN is the highest. If the word embedding dimensions are too small, the propagation of node information in the graph will be affected. On the contrary, too large word embedding dimensions will reduce the difference between feature words' embedding features, which will reduce the accuracy of SHGCN.

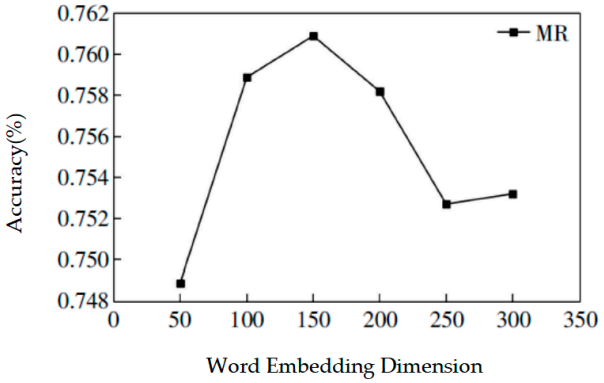

Figure 3. Accuracies of SHGCN with different embedding dimensions.

To verify the impact of the learning rate on the accuracy of the test set, this paper tested the classification results of the AGNews dataset under different learning rates. As shown in Table 2, when the learning rate decreases, the classification accuracy of the test set increases while the running time of the model increases.

Table 2. Comparison of accuracy under different learning rates.

	10^{-1}	10^{-2}	10^{-3}	10^{-5}
AGNews	83.79	86.51	88.38	88.39

Through experimental analysis, the word embedding dimension was set to 150, the learning rate was 0.001, the sliding window size was 20, and the dropout rate was 0.5.

Classification accuracy was used to measure the performance of all models. Each dataset was subjected to 50 experiments under each model, and the experimental results of seven models on three datasets are shown in Table 3.

Table 3. Comparison of accuracy of different models.

Model	AGNews		R52		MR		Mean Value
	Accuracy	Standard Deviation	Accuracy	Standard Deviation	Accuracy	Standard Deviation	
TF-IDF + LR	87.50	0.42	88.77	0.76	74.34	0.91	83.54
CNN	85.88	0.28	90.81	0.17	73.84	0.30	83.51
BiLSTM	81.91	1.42	86.12	0.68	71.76	1.41	79.93
fastText	76.62	1.74	90.34	0.17	69.09	1.84	78.69
TextGCN	79.26	7.81	92.81	0.61	76.54	0.16	82.87
TensorGCN	87.05	0.64	94.81	0.29	77.84	0.41	86.57
SHGCN	88.38	0.93	93.87	0.30	82.87	0.11	88.37

As shown in Table 3, on the AGNews and R52 datasets, the performance of the graph-based models is superior to that of the traditional CNN and BiLSTM models. This is because the graph structure allows information transfer between different types of neighbor nodes, which enables nodes to aggregate more information for feature representation. In addition, the word co-occurrence feature between words is used as the weight of edges for global sharing, which has more advantages than the local information sharing of traditional models. On the MR dataset, the accuracy of the TextGCN model is lower than that of the CNN and BiLSTM models. The main reason is that TextGCN ignores the role of word order in emotion classification, while CNN and BiLSTM construct continuous word sequences. In addition, compared with other datasets, the text of the MR dataset is too short, resulting in a smaller text graph being formed, which restricts the transmission of information between nodes.

SHGCN has the highest average accuracy on three datasets, and its accuracy on both the AGNews and MR datasets is also higher than that of other models. The accuracy of SHGCN on the R52 dataset is slightly lower than that of the TensorGCN model. The text length of the R52 dataset is relatively long, and since the TensorGCN model incorporates syntactic features, it is more suitable for datasets with long text lengths. The average accuracy of SHGCN is 5.5% and 1.8% higher than that of the TextGCN and the TensorGCN models. SHGCN integrates the entity information corresponding to a word in the heterogeneous graph and transmits the entity information to the adjacent document nodes and word nodes through the graph convolution neural network, which enriches the semantic expression of the document and word nodes. At the same time, the introduction of entity information alleviates the problem of word ambiguity, so it helps to obtain more accurate expressions. In addition, SHGCN represents the features of words and document nodes through BiLSTM, which can better capture the context semantic information of documents and has achieved good results in classification tasks that rely on word orders.

Due to the utilization of the correlation between nodes and the use of a bidirectional long short-term memory network to further explore the semantic features of text context, SHGCN achieves good results in experiments, as shown in Tables 4 and 5. In experiments on three typical datasets, compared with traditional machine learning models and mainstream deep learning models, our model achieves the best accuracy, the best precision, and the best recall rate, which verifies the practicality of our model.

Table 4. Comparison of precision under different models.

Model	AGNews		R52		MR		Mean Value
	Precision	Standard Deviation	Precision	Standard Deviation	Precision	Standard Deviation	
TF-IDF + LR	87.56	0.36	88.57	0.75	74.34	0.90	83.62
CNN	86.06	0.20	90.08	0.71	73.97	0.27	83.37
BiLSTM	82.85	0.79	85.44	0.75	72.67	0.65	80.32
fastText	77.47	1.70	90.26	0.32	69.50	1.20	79.08
TextGCN	79.09	7.79	90.62	0.67	76.55	0.16	82.09
TensorGCN	86.77	0.85	92.68	0.38	77.83	0.40	85.76
SHGCN	87.80	0.94	91.99	1.19	83.85	2.87	87.88

Table 5. Comparison of recall under different models.

Model	AGNews		R52		MR		Mean Value
	Recall	Standard Deviation	Recall	Standard Deviation	Recall	Standard Deviation	
TF-IDF + LR	87.20	0.32	88.77	0.76	74.34	0.90	83.50
CNN	85.88	0.28	90.81	0.17	73.84	0.30	83.51
BiLSTM	81.91	1.42	86.12	0.68	71.76	1.41	79.93
fastText	76.62	1.74	90.34	0.17	71.76	5.86	79.57
TextGCN	79.15	7.62	90.94	0.28	76.54	0.16	82.21
TensorGCN	86.94	0.82	94.81	0.29	77.84	0.41	86.53
SHGCN	87.63	0.97	92.76	0.69	83.81	2.90	88.07

4.5. Ablation Experiment

In order to verify the effectiveness of each part of this model, we conducted an ablation experiment. The experimental results are shown in Table 6, where w/o represents "not included". The experimental settings are described as follows:

Table 6. Comparison of accuracy under different models.

Model	AGNews	R52	MR
SHGCN	88.38	93.87	82.87
w/o BERT	88.12	93.61	75.63
w/o Entity	86.31	93.75	81.03
w/o BERT w/o Entity	86.75	93.56	76.74
w/o GCN	86.75	91.12	81.57

W/o BERT: This indicates that the BERT pre-training model is not used for feature fusion. Further, in w/o BERT, entity information is introduced to establish a text–entity–word heterogeneous graph to learn the representation of text nodes for short-text classification.

W/o Entity: w/o Entity indicates that no entity knowledge is introduced to expand short text, and feature fusion is performed on the document nodes and the word nodes learned from graph convolution networks through BERT.

W/o GCN: w/o GCN indicates that the model only encodes the input information through BERT and then outputs the text classification result.

According to Table 6, the results on the AGNews dataset indicate that the accuracy of the model without GCN can only reach 86.75%, and the accuracy of the model without BERT can only reach 88.12%. The results on the R52 dataset indicate that the accuracy of the model without GCN can only reach 91.12%, and the accuracy of the model without BERT can only reach 93.87%. These results are obviously lower than those for SHGCN. If the relationship between documents and words is not modeled, the accuracy of the model is

obviously lower than that of SHGCN, which indicates that mining the relationship between documents and words is effective for mining the features of short text.

5. Conclusions

In this study, we propose a heterogeneous graph-convolution-network-based short-text classification mode named SHGCN. It establishes a document–entity–word heterogeneous graph, which learns the features of document nodes and word nodes by capturing global information, and then feeds the learned feature representations into BiLSTM to obtain context semantic information for classification. Our experiments on three short-text baseline datasets show that the classification performance of the proposed model is superior to that of most current existing models.

Author Contributions: Conceptualization, J.H.; software, D.S. and J.W.; validation, Y.H., J.W., S.F. and Z.W.; writing—review and editing, J.H. and J.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Special Project on Network Security and Information Construction of Tianjin Normal University (52WT2329).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Wang, F.; Wang, Z.Y.; Li, Z.J. Concept-based short text classification and ranking. In Proceedings of the 23rd ACM International Conference on Information and Knowledge Management, Shanghai, China, 3–7 November 2014.
2. Yu, C.; Hua, X. SATNet: Symmetric adversarial transfer network based on two-level alignment strategy towards cross-domain sentiment classification. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 13763–13764.
3. Li, X.F.; Zhao, L.L.; He, H.B.; Li, F. Using Logistic regression model for Chinese text categorization. *Comput. Eng. Appl.* **2009**, *45*, 152–154.
4. Chang, C.C.; Lin, C.J. Libsvm. *ACM Trans. Intell. Syst. Technol.* **2011**, *2*, 1–27. [CrossRef]
5. Kim, Y. Convolutional neural networks for sentence classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014.
6. Harris, Z.S. *Distributional Structure*; Reidel Publishing Company: Dordrecht, The Netherlands, 1981; pp. 136–156.
7. Wang, S.I.; Manning, C.D. Baselines and bigrams: Simple, good sentiment and topic classification. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, Jeju, Republic of Korea, 8–14 July 2012.
8. Wallach, H.M. Topic modeling: Beyond bag-of-words. In Proceedings of the 23rd International Conference on Machine Learning-ICML'06, Pittsburgh, PA, USA, 25–29 June 2006.
9. Friedman, J.H. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* **2001**, *29*, 1189–1232. [CrossRef]
10. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
11. Zeng, J.C.; Li, J.; Song, Y.; Gao, C.; Lyu, M.R.; King, I. Topic memory networks for short text classification. *arXiv* **2018**, arXiv:1809.03664.
12. Li, M.C.; Clinton, G.; Miao, Y.J.; Gao, F. Short text classification via knowledge powered attention with similarity matrix based CNN. *arXiv* **2021**, arXiv:2002.03350.
13. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2017**, arXiv:1609.02907.
14. Yao, L.; Mao, C.S.; Luo, Y. Graph convolutional networks for text classification. *Proc. AAAI Conf. Artif. Intell.* **2019**, *33*, 7370–7377. [CrossRef]
15. Liu, X.E.; You, X.X.; Zhang, X.; Wu, J.; Lv, P. Tensor graph convolutional net- works for text classification. In Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020.
16. Yang, T.; Hu, L.; Shi, C.; Ji, H.; Li, X.; Nie, L. HGAT: Heterogeneous graph attention networks for semi-supervised short text classification. *ACM Trans. Inf. Syst.* **2021**, *39*, 1–29. [CrossRef]
17. Piao, Y.; Lee, S.; Lee, D.; Kim, S. Sparse structure learning via graph neural networks for inductive document classification. In Proceedings of the 34th AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–14 February 2020; p. 11165.
18. Xiang, L.; Zhiyuan, M.; Wei, W.; Shiyang, H. BAG: Text Classification Based on Attention Mechanism Combining BERT and GCN. *Softw. Eng. Appl.* **2023**, *12*, 230–241.

19. You, B.; Li, X.; Yao, J.; Feng, S. A Semi-supervised Short Text Classification Based on Multi-grained Graphs and Attention Mechanism. *Comput. Eng.* **2023**, *8*, 31. [CrossRef]
20. Yang, S.G.; Liu, Y.G. Short text classification method by fusing corpus features and graph attention network. *J. Comput. Appl.* **2022**, *5*, 1324–1329.
21. Linmei, H.; Yang, T.; Shi, C.; Ji, H.; Li, X. Heterogeneous Graph Attention Networks for Semi-supervised Short Text Classification. In Proceedings of the Empirical Methods in Natural Language Processing Association for Computational Linguistics, Florence, Italy, 28 July 2019.
22. Wang, S.R.; Zhou, W.A.; Jiang, C. A survey of word embeddings based on deep learning. *Computing* **2020**, *102*, 717–740. [CrossRef]
23. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
24. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. *arXiv* **2013**, arXiv:1310.4546.
25. Pennington, J.; Socher, R.; Manning, C. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014.
26. Joulin, A.; Grave, E.; Bojanowski, P.; Mikolov, T. Bag of tricks for efficient text classification. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Valencia, Spain, 3–7 April 2017.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.