IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 27, 2022, accepted February 13, 2022, date of publication February 24, 2022, date of current version March 7, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3154107

Sentiment Analysis of Weibo Comments Based on Graph Neural Network

YAN LI [1] AND NIANFENG LI [2]
[1]College of Cyber Security, Changchun University, Changchun 130022, China
[2]College of Computer Science and Technology, Changchun University, Changchun 130022, China

Corresponding author: Nianfeng Li (linianfengccu@163.com)

ABSTRACT Weibo is one of the most important online social platforms. Currently, user comments are increasing rapidly, which makes data management difficult. Comments show the non-standardized and colloquial form of expression. Traditional sentiment analysis techniques are no longer applicable to unspecified sentence analysis tasks. To mitigate overreliance on text sequences, ignoring syntactic structure, and the poor interpretability of feature space that are typical of traditional classification models, a sentiment classification model based on a graph neural network (GNN) is developed in this study. For each comment text, the dependency syntax is used to construct the semantic graph of the short text. Aiming at the heterogeneity of the semantic graph, the spatial domain graph filter is designed for feature extraction. Concurrently, long short-term memory (LSTM) is used as a state updater to filter node noise. In this method, a graph neural network is used as a semantic parser to encode the syntactic dependency tree, which can extract the semantic and syntactic features of sentences concurrently. Experimental results show that GNN-LSTM has achieved superior performance in the Weibo comments dataset by achieving 95.25% accuracy and 95.22% F1 score.

INDEX TERMS Sentiment analysis, dependent syntax, long short-term memory, graph neural network.

I. INTRODUCTION

Sentiment analysis is one of the primary tasks of natural language processing and requires identifying the trend of sentiment of users toward a given piece of text. Sentiment analysis has become an important analysis tool for the many user comments that are generated in online social platforms. Sentiment analysis algorithms based on deep learning are currently popular, and the deep learning algorithms proposed for different levels of tasks can achieve good results [1]. Recently, user comments have been rapidly increasing on online social platforms. Mining thematic views on massive texts and identifying their sentiment tendencies can provide a clear understanding of users' opinions for platforms, which helps platforms interact with users and optimize promotion strategies; this process also helps governments understand public opinion trends and make management decisions. Therefore, the task of sentiment analysis for users' short comments has become urgent. However, the widespread use of social platforms has accelerated the spoken expression of Internet phrases, and traditional sentiment analysis techniques are no longer applicable to the current non-standardized sentence analysis tasks. Traditional classifiers rely on text sequence representation and ignore syntactic components, and thereby construct a poorly interpretable feature space with Internet phrases. Models thus fail to achieve better optimal results when extracting continuous features, reducing classification performance.

To address this problem, this study develops and tests a text sentiment analysis method based on graph neural network and long short-term memory (GNN-LSTM). The semantic rules and structural dependencies of short texts are effectively preserved through syntactic analysis, and a GNN semantic parser is used to complete unstructured feature extraction. The fusion method applies a syntactic tree and GNN to sentiment analysis, which provides some theoretical and practical value when improving the performance of short text sentiment classification tasks.

The primary contributions and innovation of this paper are as follows:

1) We construct short text semantic graphs as the direct input to the graph neural network using syntactic dependency trees. The text classification task is thereby transformed into the graph classification task. Based on our knowledge, this

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

method has rarely been applied to sentiment analysis, particularly for Chinese comment analysis tasks.

2) We propose a GNN-based classification model for sentiment analysis of Weibo comments. This model explores the effectiveness of the GNN semantic parser and LSTM update and constructs an interpretable feature space. We study the classification performance of the model on a large set of Weibo comments. Additionally, the optimization strategy to mitigate oversmoothing is explored.

3) We analyze the performance of the GNN-LSTM model on a Weibo comment dataset. In addition, comparative experimental results show that GNN-LSTM also achieves good performance when oriented to comments from other platforms, demonstrating that the model has good generalizability and robustness on open online comment areas.

II. RELATED WORK

A. THE SENTIMENT ANALYSIS ALGORITHMS BASED ON DEEP LEARNING

Machine learning [2], [3] has more objective results compared to the construction of sentiment dictionaries [4], reduces the need for human resources, and solves problems such as difficulties in updating the corpus. However, it is more demanding when constructing datasets.

To address the shortcomings of machine learning algorithms, the nonlinear hierarchical neural network is used to approximate the complex function representation and learn the deep features of the corpus. Online texts can thus be processed quickly and accurately by neural networks. Therefore, deep learning methods based on neural networks are the primary research direction of current sentiment analysis [5]–[8]. As time series models, Recurrent neural networks (RNNs) are widely used in natural language processing. The unique structure of the model, which is suitable for processing context dependencies, improves classification performance over traditional machine learning algorithms [9]. However, the sequential model easily generates a large amount of redundant information and results in gradient explosion or disappearance problems. Researchers have introduced the threshold mechanism and memory unit based on RNN and proposed the LSTM model to mitigate these issues. By controlling the parameters that determine both retained and forgotten information, longer distance dependencies can be better captured. In the literature [10]–[12], the sentiment classification method based on LSTM has been proposed, and experimental results show that the LSTM model can achieve a classification accuracy of more than 90% by itself, which markedly exceeds the performance of the support vector machine (SVM). However, the LSTM model can only encode one-way sequence information. To capture bidirectional semantic dependencies, the bidirectional long short-term memory (Bi-LSTM) method was developed by combining forward and backward LSTMs to obtain the splicing vector to obtain all the information on the context. In the literature [13], [14], word-level and sentence-level features are extracted through Bi-LSTM to synthesize local sentiment in a recursive order to obtain sentiment labels of the entire text, which can capture the feature differences between sentiment polarities more accurately. However, due to the inadequacy of Bi-LSTM in capturing local semantic features, researchers have introduced a combination of Convolutional Neural Network (CNN) [15] and attention mechanisms [16]–[18] to facilitate better extraction of effective focused information [19]. In the literature [20], CNNs have been used to receive parallelized input information and combine them with multiple attention mechanisms, which effectively compensate for the overreliance on content-level attention mechanisms and identify the sentiment polarity of different targets. The literature [21]–[25] also combines improved Bi-LSTM and CNN models to capture long-term dependencies with CNNs using RNNs, obtaining better classification results.

These studies perform well on text sentiment classification tasks. However, the models only rely on sequence representations from the perspective of Euclidean data structures, ignoring the dependencies between sentence components. Thus, the constructed feature spaces are not interpretable. In response to the black-box characteristics of neural networks, new neural network models have been proposed [26], but the interpretability of each layer remains difficult to manage during model construction. To address these problems in neural network models, researchers have introduced graph neural networks (GNNs) as syntactic structure encoders in semantic analysis, providing a new research perspective to improve the performance of text sentiment analysis.

B. GRAPH NEURAL NETWORK

The application of GNNs to graph classification tasks [27] has inspired researchers. The GNN can be extended to text classification tasks by transforming texts into graphs through dependency syntax and then introducing a graph neural network as novel semantic encoder. The GNN can effectively manage complex relational structures of text and preserve global information in feature embeddings. Diego and Ivan proposed a semantic role annotation algorithm based on a graph convolutional network (GCN) that uses a GCN to encode syntactic dependency trees to produce potential feature representations of words in sentences. This study demonstrates the utility of GCN in NLP for the first time [28]. Liang Yao and Chengsheng applied a GCN to text classification tasks and constructed a corpus as a large graph containing word nodes and document nodes using GCN to capture higher-order neighborhood information. The text classification problem was thus transformed into a node classification problem, and better classification performance was obtained even with only a few annotations [29]. Due to the shortcomings of existing neural network models that ignore syntactic representation, T.H. Nguyen proposed a GCN event detection model based on syntactic dependency trees and a new pooling method to achieve optimal results [30]. For this type of problem, Shucheng Li and Lingfei Wu also proposed a neural semantic parser called Graph2Tree based on the graph neural

TABLE 1. The comparison of proposed model with others. The comparison includes six aspects, such as input forms, the required difficulty of dataset construction and the difficulty of corpus updating and so on.

	Sentiment dictionaries	Machine learning	Sequence Model	Proposed Method
Input	Embedding sequence	Embedding sequence	Embedding sequence	Semantic graph
Extracted features	Semantic features	Semantic features	Semantic features	Semantic features and structural features
Price of mark	Higher	Lower	Lower	Lower
Dataset requirements	Less	More	Less	Less
Corpus update	More difficult	Easier	Easier	Easier
Interpretability	Worse	Worse	Worse	Better

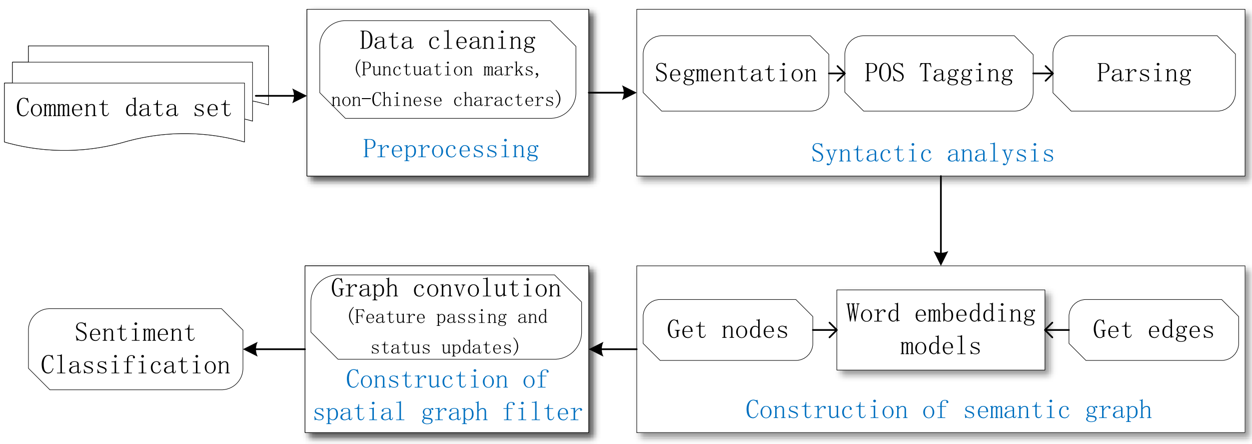

FIGURE 1. The model of sentiment analysis. It mainly includes preprocessing, syntactic analysis, construction of semantic graph and spatial domain graph filter.

network, which consisted of a graph encoder and hierarchical tree parser, and achieved remarkable results [31].

These studies effectively demonstrate the feasibility of GNNs as semantic parsers applied to NLP tasks. Currently, based on previous studies, some researchers have extended the combination of dependency syntax and GNN to sentiment analysis [32]. Shuncheng Yang and Yan Li combined topic-specific targets and proposed the Weibo stance detection method based on GCN and Bi-LSTM, which effectively improves detection accuracy [33]. However, existing research based on GNN of sentiment analysis still has a wide space for exploration and must be extended, particularly with regard to the analysis of Chinese comments.

Although the GNN as a semantic parser optimizes the deficiency in the sequence model, there are still some problems to be optimized, including the following:

1) The feature initialization of some models still depends on the sequence vector. The dependent syntax is only introduced as external knowledge to expand the feature space, but the dependency relations between sentence components cannot be directly extracted and used;

2) For the characteristics of personalized and colloquial online language, the sentence-level syntactic relationship tree has not been established. The lack of supplementation for other feature information affects classification performance;

3) The introduction of graph data typically includes a lot of noise, which may adversely affect model performance.

To address these issues, a sentiment analysis model based on GNN-LSTM is proposed. The statement of this method compared with other methods is described in Table 1. The advantage of the proposed method is that it can simultaneously extract semantic features and syntactic features to construct an interpretable feature space. First, the short review is transformed into the syntactic tree. Distinct from existing methods, we process each comment individually and transform each sentence into a semantic graph with explicit relationships instead of constructing the entire corpus as a large connected graph. Then, graph convolution operations are performed on the semantic graph to extract abstract semantic features. Finally, the feature representation of the entire graph is read out and used for sentiment classification of the short texts.

III. METHOD FOR SENTIMENT ANALYSIS BASED ON GNN-LSTM

The sentiment classification model proposed in this paper is shown in Fig. 1 and contains four primary modules:

preprocessing, syntactic analysis, semantic graph construction, and spatial graph filter construction.

The specific implementation steps are as follows:

1) First, the dataset is preprocessed, preserving valid characters.

2) Second, syntactic analysis is performed, including the three steps of word segmentation, tagging and dependency parsing.

3) Word2Vec is then used to embed subword nodes and dependent edges, and the semantic graph is constructed.

4) Then, the spatial graph filter is designed, the GNN-LSTM model is constructed, and the aggregation of features and updated states are completed.

5) The task of sentiment classification uses the softmax classifier. Finally, the model is tested to complete the performance evaluation.

A. SEMANTIC GRAPH CONSTRUCTION

Syntactic analysis analyzes grammatical rules and sentence constituents for a corpus. Phrase structure syntax and dependent syntax are the grammatical systems that are commonly used in a corpus. Dependent syntax is a theory of sentence syntax created by the French linguist L. Tesniere. Dependency syntax can transform short texts into syntactic dependency trees and simplify sentence expressions. The syntactic analysis module of the Language Technology Platform (LTP) designed by the Harbin Institute of Technology uses the dependency syntax system, which can manage 14 types of dependency relations shown in Table 2. LTP can also perform sentence component annotation and relation annotation.

TABLE 2. The partial dependency relations of dependency syntax.

Type of relationship	Symbols
主谓(subject-verb)	SBV
动宾(verb-object)	VOB
间宾(indirect-object)	IOB
前置宾语(fronting-object)	FOB

The construction of the semantic graph applies the dependency syntax by labeling the dominant subordination relationships between sentence components and transforming a short text into a tree topology. The nodes of the semantic graph are each segmented, and the root node is the subject verb, which is not controlled by other components and dominates all the remaining components. If there is a dependency relationship between the components, the edge is created between two nodes, and the relationship attribute is given. For example, for “LPT 提供了一系列中文自然语言处理工具 (LTP provides a series of Chinese natural language processing tools)”, the annotated syntactic dependencies and the transformed tree topology are shown in Fig. 2.

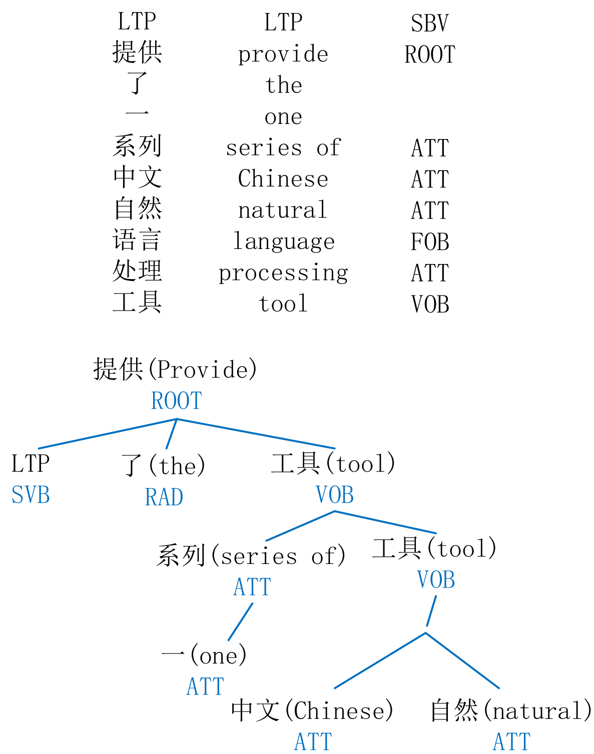

FIGURE 2. The result of syntactic analysis. Take the sentence “LPT 提供了一 系列中文自然语言 处理工具 (LTP provides a series of chinese natural language processing tools)” as example.

The process of converting a text to a semantic graph is as follows:

1) The special characters are filtered, and textual information is retained.

2) The PYLTP module is introduced, completing word segmentation, tagging and dependent syntactic analysis.

3) After obtaining the segmentation and syntactic relations, semantic graph construction is performed. The segmentation is considered as nodes, and dependency edges are established. The attributes of edges are assigned to 14 dependencies. Algorithm 1 describes this process in detail.

4) Word2Vec is used to embed nodes and edge attributes.

The sentence-level sentiment classification task is transformed into the graph classification task by converting a short text into a semantic graph.

B. GRAPH NEURAL NETWORK

A GNN is a connectedness model that captures graph relationships through message passing between nodes. Compared with traditional neural networks, GNNs can aggregate information from arbitrarily deep adjacent nodes around a node [34]. A GNN can be considered to be an extension of CNN from Euclidean data to non-Euclidean data. Using two methods (a spatial domain filter and a spectral domain filter), the GNN can effectively extracts the spatial features of topological graphs and extend the CNN to manage non-Euclidean

Algorithm 1 Semantic Graph Construction

Input:
sentences $S = \{W_1, W_2, \ldots\ldots W_i \ldots\ldots W_k\}$,
dependency syntax tree T, dependency relations R.
Output:
semantic graph G.
1: Use the verb as the Root.
2: **for** i to k **do**
3: W_i is as $node_i$
4: **for** j=1 to k and j !=i **do**
5: **if** $W_j \rightarrow$ W$_i$ **then**
6: Connect $node_i$ and $node_j \rightarrow e_{i,j}$
7: $R_{i,j} \rightarrow$ attribute($e_{i,j}$)
8: **else if** $W_i \rightarrow$ W$_j$ **then**
9: Connect $node_j$ and $node_i \rightarrow e_{j,i}$
10: $R_{j,i} \rightarrow$ attribute($e_{j,i}$)
11: **end if**
12: **end for**
13: **end for**
14: **return** G

data. The spatial domain filter is more intuitive than the spectral domain filter. The overall features of the graph can be obtained by iteratively aggregating the adjacent features of each vertex. Also, the spectral domain filter implements convolution operations on topological graphs based on spectrum theory and computes the eigenvectors of graphs with the help of the graph Laplacian matrix to complete normalization. In practical applications, the spectral domain filter is restricted to fixed connected graph processing. Therefore, the construction of a semantic graph is required. In contrast, the graph convolution operation of the spatial domain filter is more flexible and has a wider range of application scenarios. Through the graph convolution operation, the GNN can obtain all node embedding representations. The embedding representation of the entire graph can be obtained by a pooling operation called readout. The graph classification task can be accomplished by graph-level representation [35].

The graph filter is designed to aggregate the features of nodes and edges in the semantic graph and extract semantic and structural features from a short text. In this paper, syntactic trees are constructed independently for each data, with different features such as the number of nodes and edges. They are heterogeneous graphs with more complex structures and are suitable for flexible spatial domain graph convolution operations. Therefore, the spatial domain graph filter is designed to complete the graph convolution on the semantic graph.

C. MPNN FRAMEWORK

A Message Passing Neural Net (MPNN) is a formal framework of spatial domain convolution. An MPNN is not affected by graph isomorphism and places the complex and intractable spectral domain convolution under another intuitive and common methodological perspective, which increases the flexibility of application scenarios. The MPNN summarizes the spatial graph convolution into two steps, as shown in Fig. 3: message passing and state updating [36].

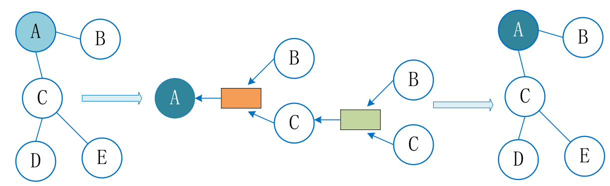

FIGURE 3. Spatail graph convolution. It describes the aggregation of adjacent feature and status update for node A.

$$H\left(X_i{}^t\right) = \rho_{j\in N(j)} M_t\left(X_i{}^t, X_j{}^t\right) \quad (1)$$

where M_t is the message delivery function, and $N_{(i)}$ is the source node, and $N_{(j)}$ is the adjacent nodes of $N_{(i)}$. t is the time step, and ρ is the method of aggregating, such as Max, Add and Average.

$$X_i{}^{t+1} = U_t(X_i^t, H(X_i^t)) \quad (2)$$

where U_t is the update function, and U_t and M_t are both differentiable functions.

The readout is calculated as follows:

$$L = R(X_i | X_i \in G) \quad (3)$$

R-functions are differentiable functions that are used to aggregate node states and must be independent of node alignment. G is the graph data, and X_i is the iterative state of N_i.

D. SPATIAL GRAPH FILTER

For heterogeneous semantic graphs, the graph convolution operation on the spatial domain is used. A spatail domain graph filter based on the MPNN framework is designed. The graph filter of the GNN-LSTM model is set as follows:

$$X_i{}^{t+1} = L^t X_i = L X_i{}^t \quad (4)$$

where X_t is the state of the node at t iterations, and L_t is the graph filter that aggregates t-th-order features. The graph filter still retains the parameter sharing property of convolution in the non-Euclidean space. The sharing mechanism of graph convolution parameters is shown in Fig. 4.

To capture the semantic information and dependencies of nodes, 14 dependency encodings are introduced as inputs to the graph filter as edge feature weights. In addition, the state of the source node depends on the states of K-group adjacent nodes, and the feature transfers of K-order adjacent nodes can be completed by K iterations. In general, a better effect of feature extraction can be achieved when the K value is taken from 1 to 2. In this study, after the comparison of experimental results, K is set equal to 3.

Due to the complex graph structure, the syntactic tree is introduced with a large amount of noise information, which affects model performance. To address this issue, the filter

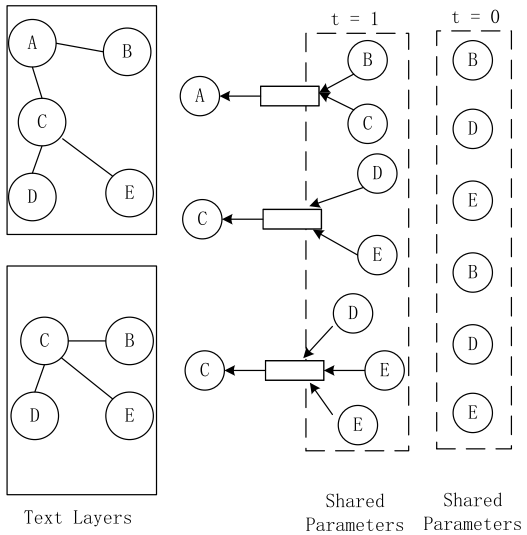

FIGURE 4. **Parameters sharing mechanism. Nodes with the same depth enjoy the shared convolution parameters.**

must retain the serialized feature analysis to optimize the model performance.

Eq. (1) and (2) are thus rewritten to propose the spatial domain graph filter of GNN-LSTM:

$$X_i^{t+1} = LSTM_t(X_i{}^t, Mean_{j\in N(j)}\phi t(|X_i^t, X_j^t|, e_{ij})) \tag{5}$$

where e_{ij} is the weight vector of the dependent edge between $N_{(i)}$ and $N_{(j)}$; ij represents the existence of connectivity between $N_{(i)}$ and $N_{(j)}$; "Mean" indicates that the method of feature aggregation is to take the mean of adjacent features; the || operator finds the Euclidean distance; and ϕ_t is the multilayer perceptron.

1) MESSAGE PASSING

Aggregation is performed by finding the mean value of all adjacent features. The feature extraction operation uses Euclidean distances to calculate the feature differences between the source and target nodes. The attribute (dependency) of the edge between the two nodes is concatenated with the distance vector and fed into the function of message passing that obtains the delivery message at moment t.

2) STATUS UPDATE

Because the state update updates the node features from moments t to t+1, there is a temporal relationship. There is also semantic noise in the features of aggregated nodes that must be filtered. Therefore, the LSTM model is used as the update algorithm. LSTM as a sequential model has the output at moment t:

$$h_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right) * tanh\left(C_t\right) \tag{6}$$

The current output depends on the previous moment state and the current cell state, which can be accomplished by updating iterations of the state at each moment. Combining LSTM with GNN can perform a serialization operation to mitigate noise and preserve the interpretability of semantics and syntax while capturing local features and aggregating global features on the graph. Thus, the two models are complementary.

3) READOUT

The number of nodes and edges in the semantic graph transformed by each text is different; thus, the graph structure is heterogeneous, which requires the adjacency matrix to be concatenated to achieve parallelization of mini-batch during processing. The operation performed is:

$$A = \begin{bmatrix} A_1 & & \\ & \ddots & \\ & & A_N \end{bmatrix}, \quad X = \begin{bmatrix} X_1 \\ \cdots \\ X_N \end{bmatrix}, Y = \begin{bmatrix} Y_1 \\ \cdots \\ Y_N \end{bmatrix} \tag{7}$$

In each batch, the index of the graph is added to each node to distinguish the nodes in each graph:

$$Batch = [0, 0, 0, 1, 1, 2, 2 \ldots\ldots n-1, n, n]^T \tag{8}$$

Readout operations on mini-batch can be performed by global pooling layers (GPLs). Pooling methods include global_max_pool, global_add_pool and global_mean_pool, etc. The feature vector of each semantic graph is computed by indices in batch. In this paper, we choose the readout function:

$$L = \varphi\left(Mean\left(X_i\right), \quad Add\left(X_i\right) | X_i \in G\right) \tag{9}$$

To reduce the readout loss, both global_add_pool and global_mean_pool are used to read out the graph features. The global_add_pool is computed as equation (10), and global_mean_pool is computed as equation (11).

$$R_i = \sum\nolimits_{n=1}^{N_i} X_n \tag{10}$$

$$R_i = \frac{1}{N_i}\sum\nolimits_{n=1}^{N_i} X_n \tag{11}$$

E. GNN-LSTM MODEL CONSTRUCTION

The structure of the GNN-LSTM model for sentiment classification by extracting semantic and structural features is shown in Fig. 5. Assuming that the word segmentation of short text isembedding word vectors is $T = \{W_1, W_2, W_3 \ldots\ldots W_n\}$, the embedding word vector is $V = \{V(W_1), V(W_2), (W_3) \ldots\ldots V(W_i)\}$, $1 \le i \le n$, and the edge attributes $E = \{V(e_1), V(e_2), V(e_3) \ldots\ldots V(e_14)\}$, feature extraction is performed on V and E using graph filters:

$$Mi = \frac{1}{N(i)} D\phi t\left(f\left(\phi A\right)\left(\left|X_i{}^t X_j{}^t\right| \odot e_{ij}\right)\right), \quad j \in N(i) \tag{12}$$

where ϕ_A is the soft-attention mechanism that assigns the weight to the feature differences of the nodes; f is the sigmoid activation function; $\odot$ is the concatenation of node features and edge features; and $D\phi_t$ is a linear transformation and

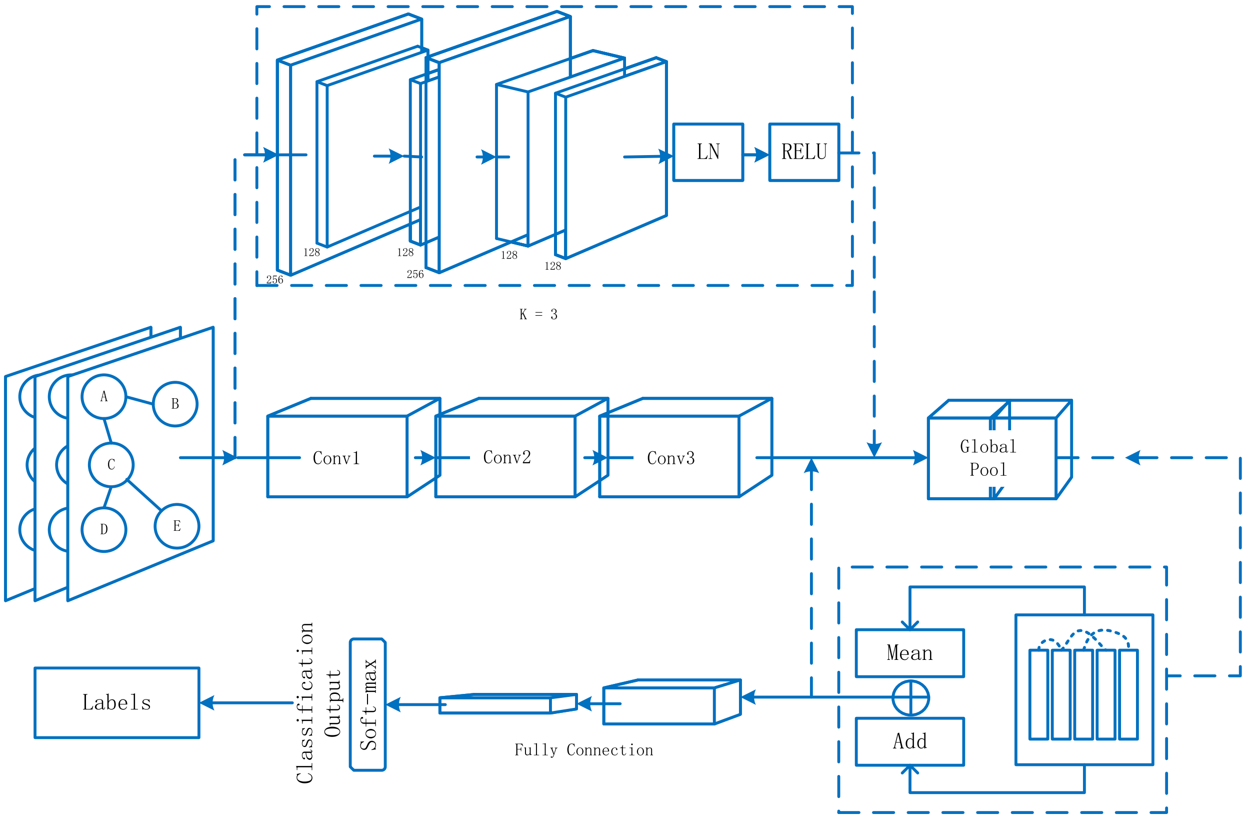

FIGURE 5. **The structure of GNN-LSTM model. It mainly includes three convolution layers with LN and RELU. The global feature is obtained by global pool function. The layer of global pool is constructed by mean function and adding function.**

ReLU activation to generalize the model. Finally, the mean value is taken for features of all adjacent nodes.

For the constructed semantic graph to reduce redundant features, no self-connected edges are added to the subword nodes. The nodes' own features are passed instead in the update stage?

$$X_i{}^{t+1} = \mathcal{F}\left(\phi_U\right) LSTM_t\left(X_i{}^t \odot M_i\right) \tag{13}$$

Combining the node's own feature with the adjacency feature as input at moment t, the LSTM model completes linear transformation and activation to obtain the state of the source node with adjacency features at moment t+1. This step completes the state update.

The graph convolution operation is then iterated three times. To stabilize the model convergence rate and stability, LayerNorm calculations are performed as Eq. (14) after each convolution layer to ensure that the features converge to the same distribution before each pass [37]:

$$X_i^{'} = \frac{X - E\left[X\right]}{\sqrt{Var\left[X\right] + \in}} + \beta \tag{14}$$

The mean and standard deviations of all nodes and channels in a mini-batch are calculated and then activated by ReLU. Finally, the aggregated 3rd-order adjacency feature is read out by the global pool algorithm and fed into the softmax function for classification.

IV. EXPERIMENTS AND ANALYSIS OF RESULTS

A. EXPERIMENTAL DATA SET

1) The Weibo_senti_100k dataset contains more than 120,000 Weibo comments, with approximately 60,000 positive and 60,000 negative comments. From this dataset, the comments are randomly divided into 70% as the training set and 15% as the test set (WB18K); then, the data are randomly divided into 70% as the training set and 10% as the test set (WB12K). Tests are performed on two test sets. The divisions of the dataset and examples are shown in Table 3.

2) The online_shopping_10_cats (Online shop) dataset has approximately 60,000 online shopping reviews containing 10 product categories, with more than 30,000 positive and negative reviews each. The comments of 10 product categories are combined. Seventy percent of the comments are randomly divided into the training set, and 15% of the comments are randomly divided into the test set. The divisions of the dataset and examples are shown in Table 4.

3) The book review dataset (Book): it has 20,000 book reviews, approximately 10,000 positive comments and 10,000 negative comments. The comments are randomly divided into 70% as the training set and 15% as test set. The divisions of the dataset and examples are shown in Table 5.

B. EVALUATION CRITERIA

To validate the model performance, accuracy, cross entropy loss function (Loss) and F1-score are selected as evaluation

TABLE 3. The example of Weibo dataset. It explains partition for training set and test set. Examples are given for training data and test data respectively.

Dataset	Partition	Training data samples
Weibo_senti_100k	70%	1)重要节日！祝快乐！(Important holiday! Happy!) 2)磕着瓜子，吃着零食，吹着徐徐凉风，看看露天电影，夏天就要这样享受！ (Knocking melon seeds, eating snacks, blowing the cool breeze and watching outdoor film, summer should be enjoyed like this!)
	80%	1)豆豆表情开始丰富多彩了。(DouDou's expressions are starting to be colorful.) 2)用心，是做菜的秘诀！(Heart is the secret of cooking!)
		Test data samples
	15%	1)凌晨三点多起床，终于把这个重要家庭成员接了回来，这下真的一家团聚啦！ (I got up at more than 3 a.m. and finally picked up this important family member. Now, it's truly a family reunion!) 2)巧克力的造型好可爱。(The shape of chocolate is so cute.)
	10%	1)所以，我们才要学会宽心，懂得宽怀。(Therefore, we should learn to be lenient and know how to be lenient.) 2)天气好到不真实，就该爬山去！(When the weather is too good to be true, it is time to go hiking!)

TABLE 4. The example of online shopping dataset. It explains partition for training set and test set. Examples are given for training and test.

Dataset	Partition	Training data samples
online_shopping_10_cats	70%	1)光盘的画面有点雪花,如果喜欢看得人,不建议购买。 (The picture of the CD is slightly blurred, if you like it, it is not recommended to buy.) 2)最近受几位老年大学学友的委托，又购买了几本《数码单反摄影轻松入门》和《RAW完全解析》，确实很不错！(Recently I commissioned by a few senior college students, and bought a few books "digital SLR photography easy to start" and "RAW complete analysis". It's so good!) 3)刚刚才拿到手，开机运行很流畅，一点卡的感觉的都没有，机身触感也很好。 (I just got it. It starts up and runs smoothly. I don't feel stuck at all. The touch of the fuselage is also very good.)
		Test data samples
	15%	1)终于认真看完了，哈哈比《男人来自火星，女人来自金星》内容更全面，文字也更有趣！既有实用性，又有可读性！ (Finally, I have read it carefully. Ha ha, it is more comprehensive and interesting than "men from Mars and women from Venus"! Both practical and readable!) 2)画面、色彩可以，就是充电好像挺久的。 (The picture and color are OK, but the charging seems to take a long time.)

TABLE 5. The example of book dataset. It explains partition for training set and test set. A certain amount of training data and test data are given.

Dataset	Partition	Training data samples
Book	70%	1)这本书写得一般，没有什么实质性的内容。(This is an average book with little substance.) 2)这本书真得很不错，对于对健康有追求的人来说，真的很有用。 (This book is truly good and really useful for people who are in pursuit of health.) 3)书拿回家两天，女儿一直粘着我给她念。她很喜欢！我也很高兴！(I got the book home for two days and my daughter has been stuck to me to read it to her. She loves it! I'm happy too!)
		Test data samples
	15%	1)确实很值的一套书，故事好看，印刷精美。推荐。 (It's a really valuable set of books with a good story and beautiful printing. Recommended.) 2)适合隔一段时间就拿出来看看，每次都会有不同的感受的，相信我。 (It's good to take it out every once in a while, and you will feel differently every time. Believe me.)

metrics and calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{15}$$

$$Loss = -\frac{1}{n}\sum\nolimits_{X}[y\ln a + (1-y)1n(1-a)] \tag{16}$$

$$F_1{-}Score = \frac{2Accuracy * Recall}{Accuracy + Recall}$$
$$(Recall = \frac{TP}{TP + FN}) \tag{17}$$

where TP (true positive) indicates the positive sentiment with correct predictions, FP (false positive) indicates the positive

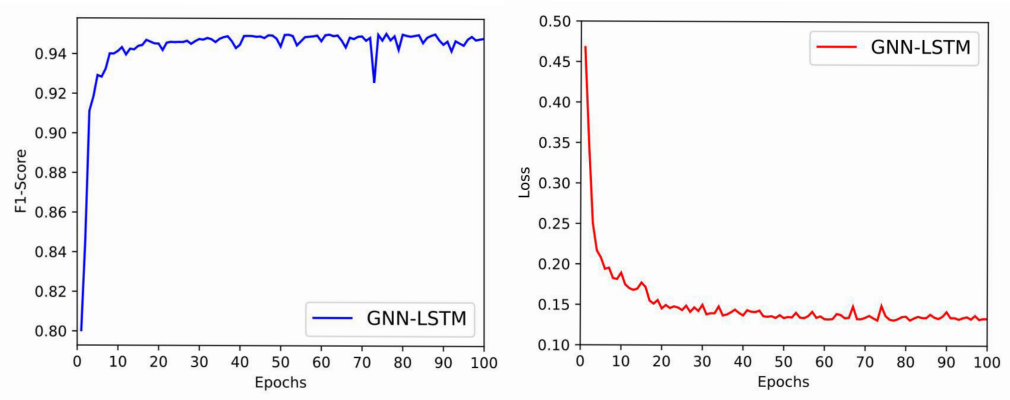

FIGURE 6. The results on validation set. During the training, the verification results are recorded. The model converges after 100 epochs.

sentiment with incorrect predictions, FN (false negative) indicates the negative sentiment with incorrect predictions, n is the total sample size, x is a specific sample data point, y is the true label, and a is the predicted output.

C. PARAMETER SETTING

The Adam optimizer and cross-entropy loss function are used for training, and the experimental results are compared after several parameter adjustments. Finally, the parameters are set to Table 6, and the sentiment classification model can determine the optimal effect.

TABLE 6. The parameters setting for experiments. During the training, the parameters setting is given.

parameters	happen to
Word vector dimension	128
Number of GNN hidden layer nodes	256
Number of LSTM hidden layer nodes	128
LOSS	CrossEntropyLoss
OPTIMZER	Adam.
LR	0.0001
BATCH-SIZE	512
WEIGHT DECAY1	0.005

During the experiment, we rearrange the dataset and randomly divide it into the training set, validation set and test set. The experiment is repeated three times, and we report the mean of the test results as the evaluation result.

Based on the parameters described above, 80% of the data from the Weibo comment dataset containing approximately 120,000 short texts are divided as the training set. Ten percent of the data are divided into the validation set, and 10% of the data are divided into the test set for testing. The validation effect of the first 100 epochs on the validation set is shown in Fig. 6.

The training curves in Fig. 6 show that the GNN-LSTM model converges quickly and achieves a high prediction accuracy. During the training process, the minimum loss obtained on the validation set is set to Best-Loss, and the corresponding parameters are saved for testing after the model converges. The results are obtained on the test set with an accuracy of 0.9485, an F1 score of 0.9480, and a loss of 0.1394.

D. COMPLEXITY AND STABILITY

We evaluate the complexity of the proposed method using 2 parameters:

1) Spatial complexity (the number of model parameters)

Table 7 shows the primary working modules of the model: graph convolution layer and the fully connected layer. The primary composition of the network layers and the output shapes are shown in table. After decomposing the model modules, we completed the statistics of the model parameters and calculated the real disk space occupied by the model. The calculation results are shown in Table 8. Additionally, we calculate the running time, taking the mean value to train one graph. Results show that the number of training parameters is approximately 1.0987MB, and the running time is approximately 0.03s, which achieves relatively balanced consumption in terms of space and time.

TABLE 7. The layer and output shape of model. The two important parts of network configuration are GNNConv module and FC layer.

GNN-LSTM	Layer(type	Output shape
GNNConv	Message passing	-1×1×256
	LSTM	-1×1×128
	Output	-1×1×128
	LN	-1×1×128
FC Layer	Linear1	-1×1×64
	Linear2	-1×1×2

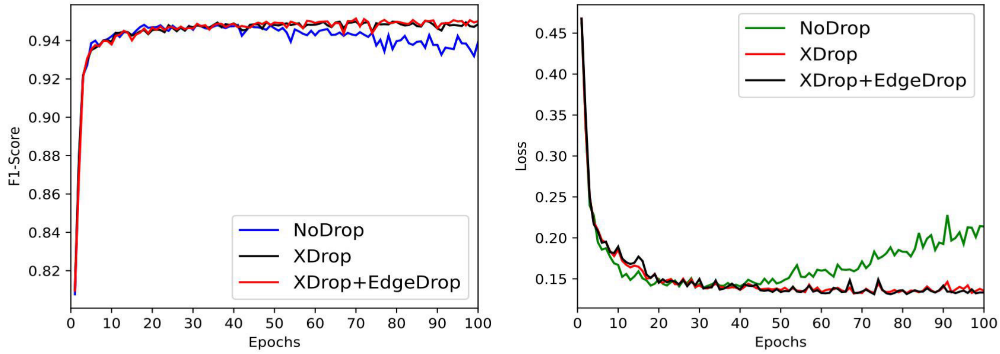

FIGURE 7. The comparison of strategy effectiveness. The validation curves with only strategy 1), strategies 1) and 2), and without optimization are drawn to compare the effectiveness of optimization strategy.

2) Time complexity (the number of model operations)

The proposed model is implemented based on the spatial domain graph filter. The computational steps are broken down into two steps: transfer and update. Feature passing is completed by a mean function including features of adjacent nodes and edges. Because the semantic graph is sparse, the index lists of nodes and edges are used for storage in real calculation. Assuming that the semantic graph has N nodes and E edges, then there are at most N-1 adjacent nodes and dependent edges of a source node because there are pointing relations of dependent syntax. Therefore, the time complexity of the transfer stage is:

$$T(n1) = N \times (N-1)(D_v + D_E) \tag{18}$$

The update uses LSTM as the updater, and the time complexity of LSTM is noted as the time complexity of the update stage. The overall complexity is obtained by adding the complexities of two computational stages:

$$T(n_2) = N \times (N-1)(D_v + D_E) + N \times (D_V + D_S) \tag{19}$$

where N is the number of nodes in the graph; D is the vector computation; D_V, D_E, and D_S are the vector computation of node features, edge features, and adjacent features, respectively.

From Eq. (19), the time complexity of the proposed model is:

$$T(n) = O(nd(n+d)) \tag{20}$$

where n is the number of nodes, and d is the vector length.

Based on this inference, we estimated the model computation in Table 8, and evaluated the model with FLOPs, which obtained a value of approximately 1.807×108.

According to this complexity analysis and the training status in Figure 6, the proposed model has a moderate number of parameters and a reasonable trainable space. The proposed model also achieves a high computational efficiency and generalizability. Therefore, the model has high stability and robustness.

TABLE 8. The statistics of model complexity. The number of parameters is evaluated for spatial complexity. And the number of model operations is evaluated for time complexity.

Time	Trainable parameters	Model size	Computation
0.03139 s	1152072 Params	4.4065 MB	180748900 FLOPs

E. OVER-SMOOTHING OPTIMIZATION

The GNN aggregates adjacent features through message passing and obtains global graph features by continuous iteration. This aggregation will eventually cause the node states to converge. Even if the graph convolution layer is added to expand the perceptual range, no more effective features can be extracted to improve model performance due to oversmoothing that is common to GNNs.

Mitigating this issue optimally requires adding noise and aggregating differentiated features to improve model generalizability during message passing. Therefore, the strategies of node-feature loss and edge-feature loss are used to mitigate oversmoothing optimally using the following strategies:

1) Strategy of node-features loss: node features extracted from each convolution layer are input to the dropout layer during message passing, losing 20% of features. This operation enhances the data, and expands the difference in node features passed by each layer.

2) Strategy of edge-features loss: all edge attributes are randomly deactivated by 50% in the input stage. This strategy performs feature transformations for 14 fixed dependencies to increase feature diversity and distinguish edge states. The features are then input to the graph filter along with the node features for message passing.

Validation results with and without optimization are shown in Fig. 7.

The curves in Fig. 7 show that the model exhibits marked overfitting when the optimization strategy is not used. After applying the optimization strategies, oversmoothing is reduced, and generalizability is enhanced. Overfitting is thus effectively prevented.

TABLE 9. **Comparison of optimization effect. The 'XDrop' column denotes strategy of node-features loss, and 'EdgeDrop' column denotes strategy of edge-fetures loss. The best result is highlighted in bold for each column.**

Strategy			ACC	F1	LOSS
XDrop	EdgeDrop	Non			
		✓	0.9485	0.9480	0.1394
✓			0.9500	0.9495	0.1320
✓	✓		**0.9525**	**0.9522**	**0.1312**

TABLE 10. **Effect of layers and updater. K denotes the number of convolution layers. The LSTM and SUM denote two types of updaters. The top performance is highlighted in bold.**

Method					ACC	F1	LOSS
K=1	K=2	K=3	LSTM	SUM			
✓			✓		0.9419	0.9429	0.1676
	✓		✓		0.9491	0.9490	0.1384
		✓	✓		**0.9525**	**0.9522**	**0.1312**
		✓		✓	0.9503	0.9504	0.1415

Test results after applying the optimization strategies are shown in Table 9. The test results after adopting strategy 1 achieved an accuracy of 0.9500, an F1 score of 0.9495, and a loss of 0.1320. Compared to the model without optimization, performance improved by approximately 0.15%. With strategy 2, the test results achieve an accuracy of 0.9525, an F1 score of 0.9522, and a loss of 0.1312. Model performance is thus improved.

F. ABLATION EXPERIMENTS

1) EFFECT OF NETWORK LAYERS AND UPDATE

Two experiments were performed as follows:

Model performance was compared when the convolution layer K was set to 1, 2 and 3.

The model performances of the GNN and GNN-LSTM were compared to demonstrate the effectiveness of the LSTM updater.

Comparative results are shown in Table 10, and the optimal classification performance is obtained when the number of layers is 3. As the state updater, LSTM can effectively remove the feature noise of the graph and optimize the effect of classification.

2) EFFECT OF NORMALIZATION

Comparison experiments were performed as follows:

The functions of normalization as BatchNorm (BN), LayerNorm (LN), GraphNorm (GN) and non-normalization were used in the normalization layers to compare classification performances.

Comparative results are shown in Table 11, and the experiments show that LN can effectively prevent overfitting and improve generalizability.

TABLE 11. **The effect of normalization. The model performance adopted three normalization methods and without normalization is compared. The best result is highlighted in bold.**

Method				ACC	F1	LOSS
BN	LN	GN	Non			
✓				0.9483	0.9484	0.1415
		✓		0.9484	0.9485	0.1411
	✓			**0.9525**	**0.9522**	**0.1312**
			✓	0.9392	0.9392	0.1735

3) EFFECT OF AGGREGATION SCHEME

Comparison experiments were performed as follows:

When implementing GNNConv based on the MPNN framework, the aggregation strategies of adding (ADD), averaging (MEAN), and maximizing (MAX) are used to explore the impact of aggregation methods on performance.

Experimental results are shown in Table 12 and indicate that the adding strategy retains feature integrity while retaining a lot of noise. The maximizing strategy may lose some effective information. In contrast, the aggregation method of averaging can effectively obtain the adjacent features and has good performance.

TABLE 12. **The effect of aggregation scheme. The scores of three aggregation strategies are calculated, and the optimal aggregation scheme is selected.**

Method			ACC	F1	LOSS
ADD	MEAN	MAX			
✓			0.9501	0.9496	0.1397
		✓	0.9513	0.9508	0.1381
	✓		**0.9525**	**0.9522**	**0.1312**

G. COMPARISON OF RELATED WORK

The benchmark methods for the comparison experiments are described as follows.

1) Long Short-Term Memory (LSTM) [38]: A variant of RNN solves the long-term dependency problem.

2) Bidirectional Long Short-Term Memory (BiLSTM) [39]: The model consists of a forward LSTM and a backward LSTM spliced together. Forward and backward hidden vectors are concatenated for sentiment classification tasks.

3) TextING [40]: This algorithm constructs a text graph by unique words as vertices and co-occurrence relations between words as edges. Feature embeddings merged with their own features are obtained by gated GNN for updating. Finally, the graph-level feature is read out.

4) TextGCN [29]: It constructs a heterogeneous graph for the entire corpus, with deduplicated words and documents as nodes. Edges are constructed between documents and words, and words and words. Document features and lexical features

are embedded using TextGCN. The node features of the last layer are used for label prediction.

5) BiLSTM-GCN [33]: Word vectors are first input into the BiLSTM model for encoding. The resulting hidden vectors are used as the initialization input to the GCN. Dependencies of distant nodes are captured by the GCN. Finally, the representation of text features is obtained by the global pooling operation for sentiment classification.

The performance of these models is compared, and results are shown in Tables 13 and 14. "WB18K" denotes the test set with 1,8000 Weibo comments; "WB12K" denotes the test set with 1,2000 Weibo comments; "Online shop" denotes the test set constructed from online_shopping_10_cats, including 9,000 comments; and "Book" denotes the test set with 3,000 comments about book reviews.

TABLE 13. The comparison of accuracy. The accuracy of proposed method is compared with benchmark methods on four test sets. The best result is highlighted in bold for each column.

	WB18k	WB12k	Online shop	Book
LSTM [38]	0.9266	0.9358	0.9054	0.7859
BiLSTM [39]	0.9279	0.9341	0.9063	0.7928
TextING [40]	0.9251	0.9262	0.8724	0.7485
TextGCN [29]	0.8737	0.8819	0.9170	**0.8239**
BiLSTM-GCN [33]	0.9087	0.9266	0.8952	0.8049
GNN-LSTM	**0.9511**	**0.9525**	**0.9196**	0.8086

TABLE 14. The comparison of f1-score. The f1-score of proposed method is compared with benchmark methods on four test sets. The best result is highlighted in bold for each column.

	WB18k	WB12k	Online shop	Book
LSTM [38]	0.9258	0.9354	0.9071	0.7898
BiLSTM [39]	0.9271	0.9335	0.9091	0.7924
TextING [40]	0.9276	0.9278	0.8680	0.7406
TextGCN [29]	0.8791	0.8849	0.9145	**0.8193**
BiLSTM-GCN [33]	0.9081	0.9281	0.8950	0.7904
GNN-LSTM	**0.9504**	**0.9522**	**0.9212**	0.8095

As shown in Tables 13 and 14, the proposed model generalizes well when oriented to comments from other platforms. The model achieves scores above 90% on both the Weibo comments and online shopping comments, and achieves the best performance on the WB12K test set (an accuracy of 95.25% accuracy and an F1 score of 95.22%). These results demonstrate that the GNN-LSTM model has good sentiment classification for short comments on Weibo and can also optimally perform for other open comment areas.

In addition, 12 sets of data were constructed on dataset 1 with a data volume of 10,000 to 120,000 in increments of 10,000. In addition, 10% of the comments were selected from each of the datasets for testing. The test results of relevant models on 12 groups of datasets with different sizes are compared, as shown in Fig. 8, to analyze the effect of data volume on model performance.

Fig. 8 shows the accuracy curves and F1-score curves of benchmark methods on 12 groups of datasets. GNN-LSTM achieves the optimal and smoothest curve in the comparison experiment, which shows that the method of constructing semantic graphs based on internal connectivity and designing a spatial graph filter based on it has a more stable performance on datasets of different sizes. GNN-LSTM still has an effective classification effect on small sample datasets.

As shown in Tables 13 and 14, the GNN-LSTM model outperforms the other models except for the Book dataset. Possible reasons for this result are as follows: GNN-LSTM constructs semantic graphs based on the dependent syntax of each short text for graph convolution and relies on internal connections of texts without establishing external connections between texts, which may affect performance. However, compared with the graph construction method of embedding words and documents throughout the corpus, the composition of GNN-LSTM is more flexible, saves a large amount of storage space, and reduces spatial complexity. Therefore, GNN-LSTM achieves better stability and generalizability.

H. ANALYSIS OF ADVANTAGES AND LIMITATIONS

Compared with benchmark methods, the proposed method has the following advantages:

1) Compared with methods in [38] and [39], an interpretable feature space is constructed, which reduces the dependence on sequences and extracts both semantic and syntactic features of sentences.

2) Compared with methods in [29] and [40], the composition method based on the entire corpus has been abandoned, and a more flexible composition method of single text is used instead to save storage space. The use of a spatial graph filter also reduces the computation.

3) Compared with the method in [33], the proposed method does not use the semantic graph as external knowledge for feature supplementation but rather completes the feature extraction directly on the semantic graph. The constructed graph filter and feature space have a better interpretation, which completes the intuitive transformation from text classification to graph classification.

However, there are several implementation challenges with the proposed method.

1) When constructing semantic graphs, only certain special characters are filtered in the preprocessing stage to perform complete dependency parsing, but stop words cannot be removed. Although complete structural features are retained, a lot of noise exists in the graph data, which may affect model performance.

2) The GNN algorithm implemented by the MPNN framework requires considering both feature passing and state updating in the optimization stage, which increases the difficulty of model optimization.

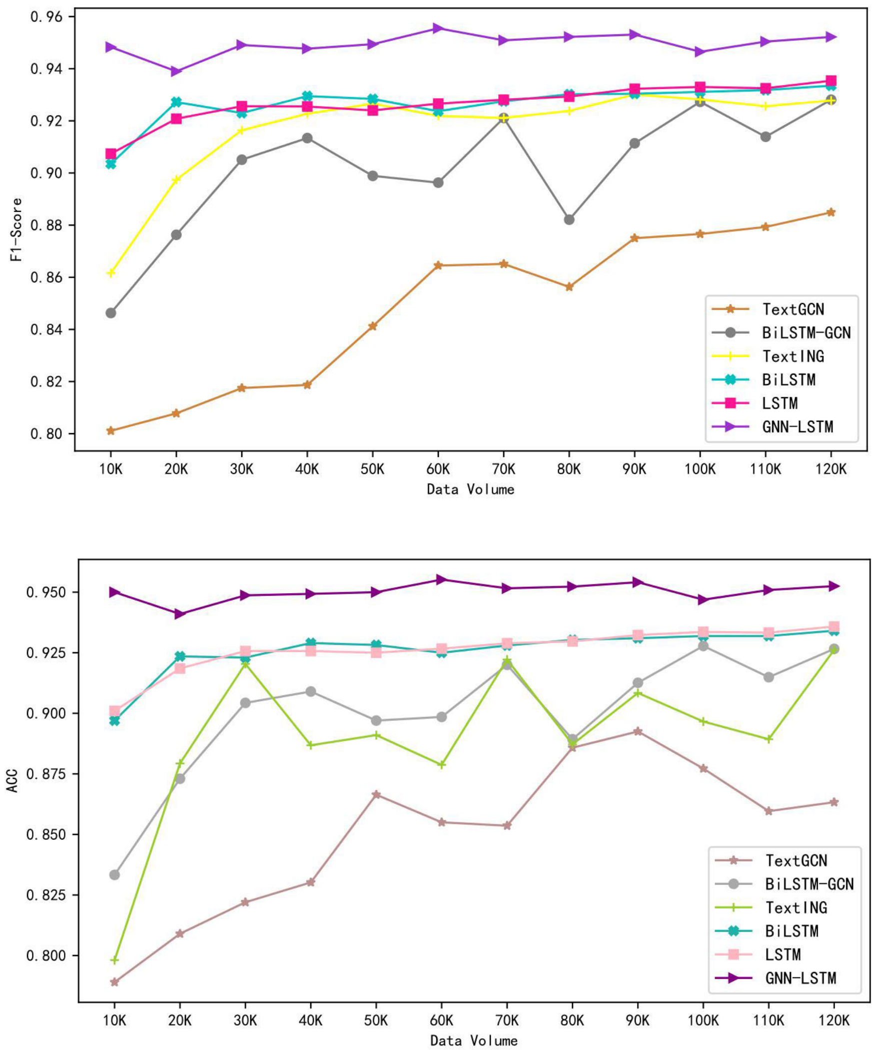

FIGURE 8. **The comparison of model accuracy. The impact of data volume on model performance is tested. The benchmark dataset is Weibo_senti_100k, and the data volume is in increment of 10,000.**

3) The method of single-text composition only describes the topological structure inside the text but fails to establish the connection between sentences and obtain the overall features of the corpus. With uneven data quality, this issue is more likely to affect model accuracy.

These three points highlight the limitations of the proposed method compared with the other methods. In future research, we plan to use optimization strategies to address these limitations.

V. CONCLUSION & FUTURE DIRECTION

Currently, the number of user comments on social platforms such as Weibo are rapidly increasing and contain spoken expressions. In this study, we develop and test a sentiment analysis model based on GNN-LSTM. While extracting semantic features, the syntactic features are retained to enhance the interpretability of the feature space. GNN-LSTM captures continuous features and optimizes the sentiment classification performance. The LSTM model filters the graph noise, and effective aggregated information is retained. Experimental results show that the GNN-LSTM model achieves good generalizability for the open comment area of the Internet and can effectively accomplish sentiment analysis with short texts.

In future work, we plan to use two strategies to optimize the proposed model's performance to address the learning limitations on heterogeneous semantic graphs due to the strong homogeneity assumption of the GNN model. These strategies are 1) constructing relationships between texts enriches the feature space, and 2) introducing higher-order adjacent features enhances the features of semantic graphs and increases the perception range of convolution.

REFERENCES

[1] B. Ghanem, P. Rosso, and F. Rangel, "An emotional analysis of false information in social media and news articles," *ACM Trans. Internet Technol.*, vol. 20, no. 2, pp. 1–18, May 2020.

[2] G. A. Ruz, P. A. Henríquez, and A. Mascareño, "Sentiment analysis of Twitter data during critical events through Bayesian networks classifiers," *Future Gener. Comput. Syst.*, vol. 106, pp. 92–104, May 2020.
[3] A. Mitra, "Sentiment analysis using machine learning approaches (Lexicon based on movie review dataset)," *J. Ubiquitous Comput. Commun. Technol.*, vol. 2, no. 3, pp. 145–152, Sep. 2020.
[4] B. R. Chakravarthi, V. Muralidaran, R. Priyadharshini, and J. P. McCrae, "Corpus creation for sentiment analysis in code-mixed Tamil-English text," 2020, *arXiv:2006.00206*.
[5] Z. X. Z, "Research on text sentiment analysis method based on neural network," M.S. thesis, Hunan Univ., Hunan, China, 2019.
[6] N. C. Dang, M. N. Moreno-García, and F. De la Prieta, "Sentiment analysis based on deep learning: A comparative study," *Electronics*, vol. 9, no. 3, p. 483, Mar. 2020.
[7] U. Naseem, I. Razzak, K. Musial, and M. Imran, "Transformer based deep intelligent contextual embedding for Twitter sentiment analysis," *Future Gener. Comput. Syst.*, vol. 113, pp. 58–69, Dec. 2020.
[8] W. Li, W. Shao, S. Ji, and E. Cambria, "BiERU: Bidirectional emotional recurrent unit for conversational sentiment analysis," *Neurocomputing*, vol. 467, pp. 73–82, Jan. 2022.
[9] L. Liu and X. Yu, "Recurrent neural network (RNN) and its application research," *Sci. Technol. Vis.*, vol. 32, no. 2, pp. 54–55, 2019.
[10] S. C. Peng, "Research on sentiment classification of product reviews based on RNN and LDA model," M.S. thesis, Zhejiang Sci-Tech Univ., Hangzhou, China, 2018.
[11] B. Ghanem, P. Rosso, and F. Rangel, "An emotional analysis of false information in social media and news articles," *ACM Trans. Internet Technol.*, vol. 20, no. 2, pp. 1–18, May 2020.
[12] K. Sangeetha and D. Prabha, "Sentiment analysis of student feedback using multi-head attention fusion model of word and context embedding for LSTM," *J. Ambient Intell. Humanized Comput.*, vol. 12, no. 3, pp. 4117–4126, Mar. 2021.
[13] Z. Zeng and P. Wan, "Analysis of microblog sentiment in public security events based on bi-attention and bi-LSTM," *Inf. Sci.*, vol. 37, no. 6, p. 7, 2019.
[14] J. Wei, J. Liao, Z. Yang, S. Wang, and Q. Zhao, "BiLSTM with multi-polarity orthogonal attention for implicit sentiment analysis," *Neurocomputing*, vol. 383, pp. 165–173, Mar. 2020.
[15] H. Sadr, M. M. Pedram, and M. Teshnehlab, "Multi-view deep network: A deep model based on learning features from heterogeneous neural networks for sentiment analysis," *IEEE Access*, vol. 8, pp. 86984–86997, 2020.
[16] T. Kim and B. Lee, "Multi-attention multimodal sentiment analysis," in *Proc. Int. Conf. Multimedia Retr.*, Jun. 2020, pp. 436–441.
[17] G. Xu, Z. Yu, Z. Chen, X. Qiu, and H. Yao, "Sensitive information topics-based sentiment analysis method for big data," *IEEE Access*, vol. 7, pp. 96177–96190, 2019.
[18] M. M. Truşcă, D. Wassenberg, F. Frasincar, and R. Dekker, "A hybrid approach for aspect-based sentiment analysis using deep contextual word embeddings and hierarchical attention," in *Proc. Int. Conf. Web Eng.* Cham, Switzerland: Springer, 2020, pp. 365–380.
[19] A. Onan, "Sentiment analysis on product reviews based on weighted word embeddings and deep neural networks," *Concurrency Comput., Pract. Exper.*, vol. 33, no. 23, p. e5909, Dec. 2021.
[20] B. Liang, Q. Liu, J. Xu, Q. Zhou, and P. Zhang, "Specific target emotion analysis based on multi-attention convolutional neural network," *J. Comput. Res. Develop.*, vol. 54, no. 8, p. 12, 2017.
[21] Y. Guo, K. Zhao, and L. Pan, "Combined improved Bi-LSTM and CNN text sentiment analysis," *Inf. Technol.*, vol. 351, no. 2, pp. 58–61, 2021.
[22] H. Zhao, L. Wang, and E. A. Wang, "Text sentiment analysis based on BiLSTM-CNN serial hybrid model," *Comput. Appl.*, vol. 40, no. 1, pp. 16–22, 2019.
[23] M. E. Basiri, S. Nemati, M. Abdar, E. Cambria, and U. R. Acharya, "ABCDM: An attention-based bidirectional CNN-RNN deep model for sentiment analysis," *Future Gener. Comput. Syst.*, vol. 115, pp. 279–294, Feb. 2021.
[24] M. Usama, B. Ahmad, E. Song, M. S. Hossain, M. Alrashoud, and G. Muhammad, "Attention-based sentiment analysis using convolutional and recurrent neural network," *Future Gener. Comput. Syst.*, vol. 113, pp. 571–578, Dec. 2020.
[25] R. K. Behera, M. Jena, S. K. Rath, and S. Misra, "Co-LSTM: Convolutional LSTM model for sentiment analysis in social big data," *Inf. Process. Manage.*, vol. 58, no. 1, Jan. 2021, Art. no. 102435.
[26] T. Ito, K. Tsubouchi, H. Sakaji, T. Yamashita, and K. Izumi, "Contextual sentiment neural network for document sentiment analysis," *Data Sci. Eng.*, vol. 5, no. 2, pp. 180–192, Jun. 2020.
[27] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.
[28] D. Marcheggiani and I. Titov, "Encoding sentences with graph convolutional networks for semantic role labeling," 2017, *arXiv:1703.04826*.
[29] L. Yao, C. Mao, and Y. Luo, "Graph convolutional networks for text classification," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, no. 1, pp. 7370–7377.
[30] T. H. Nguyen and R. Grishman, "Graph convolutional networks with argument-aware pooling for event detection," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1, pp. 5900–5907.
[31] S. Li, L. Wu, F. Xu, F. Xu, and S. Zhong, "An empirical study of graph neural networks based semantic parsing," in *Proc. 25th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Anchorage, AK, USA, Aug. 2019.
[32] K. Wang, W. Shen, Y. Yang, X. Quan, and R. Wang, "Relational graph attention network for aspect-based sentiment analysis," 2020, *arXiv:2004.12362*.
[33] S. Yang, Y. Li, and Q. Zhao, "Detection of microblog position based on GCN and bi-LSTM," *J. Chongqing Univ. Technol., Natural Sci.*, vol. 34, no. 6, pp. 167–173, 2020.
[34] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, Jan. 2020.
[35] Z. Zhang, P. Cui, and W. Zhu, "Deep learning on graphs: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 1, pp. 249–270, Jan. 2022.
[36] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1263–1272.
[37] J. Lei Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*.
[38] Y. Tao, "Research on sentiment analysis of online public opinion for emergencies," vol. 34, no. 6, pp. 167–173, 2021.
[39] M. Zou and Y. Xin, "Research on emotion analysis of movie pop-ups based on text mining," *Sci. Technol. Innov. Appl.*, vol. 11, no. 24, pp. 57–59, 2021.
[40] Y. Zhang, X. Yu, Z. Cui, S. Wu, Z. Wen, and L. Wang, "Every document owns its structure: Inductive text classification via graph neural networks," *arXiv:2004.13826*, 2020.

YAN LI was born in Shandong, China. She is currently pursuing the degree in cyber security with Changchun University, China. Her research interests include natural language processing and sentiment analysis.

NIANFENG LI has been teaching with Changchun University, since 1997. He currently works as the Dean of the School of Computer Science and Technology, Changchun University, the Director of the Jilin Computer Society, the Vice President of the Jilin Robotics Society, and the Secretary General of the Jilin Computer Education steering Committee. His research interests include artificial intelligence, computer application technology, image processing, and campus security technology.