Received July 21, 2020, accepted August 5, 2020, date of publication August 11, 2020, date of current version August 24, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3015770

An Integration Model Based on Graph Convolutional Network for Text Classification

HENGLIANG TANG[1,2], **YUAN MI**[1], **FEI XUE**[1], **AND YANG CAO**[1]

[1]School of Information, Beijing Wuzi University, Beijing 101149, China
[2]Beijing Key Laboratory of Multimedia and Intelligent Software Technology, Beijing University of Technology, Beijing 100124, China

Corresponding author: Yuan Mi (15029924182@163.com)

This work was supported in part by the Beijing Key Laboratory of Intelligent Logistics System under Grant BZ0211, in part by the Beijing Intelligent Logistics System Collaborative Innovation Center, in part by the Beijing Municipal Natural Science Foundation, in part by the Beijing Youth Top-notch Talent Plan of High-Creation Plan under Grant 2017000026833ZK25, in part by the Canal Plan-Leading Talent Project of Beijing Tongzhou District under Grant YHLB2017038, and in part by the Grass-roots Academic Team Building Project of Beijing Wuzi University under Grant 2019XJJCTD04.

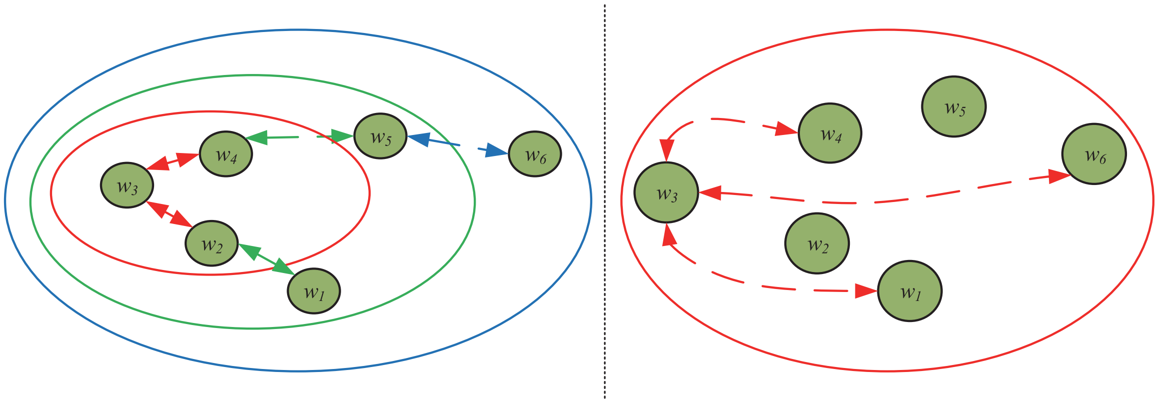
ABSTRACT Graph Convolutional Network (GCN) is extensively used in text classification tasks and performs well in the process of the non-euclidean structure data. Usually, GCN is implemented with the spatial-based method, such as Graph Attention Network (GAT). However, the current GCN-based methods still lack a more reasonable mechanism to account for the problems of contextual dependency and lexical polysemy. Therefore, an improved GCN (IGCN) is proposed to address the above problems, which introduces the Bidirectional Long Short-Term Memory (BiLSTM) Network, the Part-of-Speech (POS) information, and the dependency relationship. From a theoretical point of view, the innovation of IGCN is generalizable and straightforward: use the short-range contextual dependency and the long-range contextual dependency captured by the dependency relationship together to address the problem of contextual dependency and use a more comprehensive semantic information provided by the BiLSTM and the POS information to address the problem of lexical polysemy. What is worth mentioning, the dependency relationship is daringly transplanted from relation extraction tasks to text classification tasks to provide the graph required by IGCN. Experiments on three benchmarking datasets show that IGCN achieves competitive results compared with the other seven baseline models.

INDEX TERMS Bidirectional long short-term memory network, dependency relationship, graph convolutional network, part-of-speech information, text classification.

I. INTRODUCTION

Text classification is always a hot topic of Natural Language Processing (NLP), which is widely applied for text recognition and opinion extraction [1]–[4]. Currently, a large amount of the non-euclidean structure data, which can be quantified and analyzed, is generated by the social media every day, such as social network reviews, interview records, product reviews, email records, etc. Over the past few decades, the non-euclidean structure data has been studied mainly based on the traditional classification methods (e.g. Support Vector Machine (SVM) [5]), the typical neural network methods (e.g. Convolutional Neural Network (CNN) [6], Recurrent Neural Network (RNN) [7], Capsule Networks [8]), and their excellent variants (e.g. Bidirectional Recurrent Neural Network (BRNN) [9], Long Short-Term Memory (LSTM) [10] Network, Gated Recurrent Unit (GRU) [11], Recurrent Convolutional Neural Network (RCNN) [12], Convolutional Recurrent Neural Network (CRNN) [13]). However, these methods have been greatly challenged on the graph-structure data. For instance, the graph-structure data cannot be directly processed by CNN, for the reason that CNN cannot maintain the translation invariant. Besides, the fixed size of the convolution kernel limits the range of the dependency. Therefore, the methods based on Graph Convolutional Network (GCN) [14] receive a growing attention from researchers and engineers. With regarding the graph as a spectral graph, GCN can realize the end-to-end learning of node feature and structure feature. Moreover, GCN is applicable to the graph of arbitrary topology. Although GCN is gradually becoming a good choice for text classification based on the graph, there are still certain defects not to be neglected in current study.

The associate editor coordinating the review of this manuscript and approving it for publication was Ali Shariq Imran.



Original GCN

IGCN based on the Dependency Relationship

FIGURE 1. The difference of capturing the dependency between the original GCN and IGCN.

The original GCN cannot capture the short-range contextual dependency and the long-range contextual dependency together. Look at this example "the movie, which is the product of an unknown French director, is wonderful.". Due to the mechanism that GCN only aggregates the information of the direct neighbor nodes, GCN can only capture the short-range contextual dependency information. This question can only be solved by increasing the number of GCN layers to capture the long-range contextual dependency, such as the dependency between the words "movie" and "wonderful". However, the current researches reveal that the multi-layer GCN for text classification tasks will give rise to a high spatial complexity [15]. Meanwhile, the over-smoothing of the node feature will also be caused by increasing the number of network layers, which will make local features converge to a similar value.

In addition to the problem of contextual dependency, the problem of lexical polysemy also exists in GCN. The problem of lexical polysemy can be described that the same word may express different semantics in same or different positions. In the sentences "I bought an apple." and "I bought an apple X.", the meaning of the word "apple" is different due to the difference of the context. Meanwhile, in the sentences "Our object is to further cement trade relations." and "Their voters do not object much.", the meaning of the word "object" is also different owing to the difference of the part-of-speech. Although the relevant researches have claimed that the problem of lexical polysemy can be solved without relying on the syntactic information and the semantic information, their effects are not up to expectation unfortunately [16], [17].

To overcome the problems of contextual dependency and lexical polysemy, an improved GCN (IGCN) is proposed for text classification in this paper. Based on the original GCN, IGCN introduces the Bidirectional Long Short-Term Memory (BiLSTM) [18], [19] Network, the Part-of-Speech (POS) information, and the dependency relationship[1] reasonably.

In this paper, the text feature and the POS feature sequentially obtained through BiLSTM will be applied to solve the problem of lexical polysemy. Through constructing the dependency relationship, IGCN can take good advantage of the short-range contextual dependency and the long-range contextual dependency together. Meanwhile, the adjacency matrix based on the dependency relationship will also be generated to provide syntactic constraints. Subsequently, the different attentions of the neighbor nodes to the central node can be learned during the aggregation process. Namely, the weights of the features will be calculated by the attention mechanism in the propagation process. What is worth mentioning, to provide the graph required by IGCN, the dependency relationship is daringly transplanted from relation extraction tasks to text classification tasks. The difference of capturing the dependency between the original GCN and IGCN is shown in Fig. 1.

Experiments on three benchmarking datasets demonstrate that the problems in the current GCN-based method can be effectively addressed by IGCN which has some advantages than the other researches. The main contributions of this paper are as follows:

- The text information and the POS information can be effectively applied to generate the initial features by BiLSTM. Simultaneously, the features can not only make up for the deficiency of the content-level context effectively, but also provide a new idea to address the problem of lexical polysemy.

[1] the POS information and the dependency relationship are generated through spaCy toolkit. The POS information of each word is generated by the POS tagging. The dependency relationship between words is generated by the dependency parsing.

- The dependency relationship and the attention mechanism are fully integrated into the IGCN. They can effectively deal with the problem of contextual dependency, and reduce the number of GCN layers partly. Namely, they can solve the high space complexity and over-smoothing caused by the multi-layer GCN indirectly. Such a cross-task study is useful to meet the challenges in NLP, and further demonstrates the significance of the dependency relationship.
- A large number of experimental results not only prove the reasonability of integrating the BiLSTM, the POS information, and the dependency relationship, but also prove the efficiency of IGCN for text classification. IGCN will contribute to the continuous development of the research field of the non-euclidean structure data.

II. RELATED WORK

Unlike the traditional classification methods based on extracting text features manually, the current methods based on the deep learning can directly output the category of the text by training the neural network. For example, Tang *et al.* [20] adopted two LSTMs for text classification which effectively integrate sentiment words and contextual information. Zhang *et al.* [21] classified texts through introducing the sentiment word information into BiLSTM. Yang *et al.* [22] integrated the common sense into the deep neural networks based on BiLSTM to enhance the accuracy of text classification. Xue and Li [23] utilized CNN and the gate mechanism inversely to achieve higher accuracy which broke away from the network structure on the base of RNN and attention mechanism. Huang and Carley [24] achieved amazing results through designing the parameterized filters and the gate mechanism on CNN to capture text features. Li *et al.* [25] proposed a feature transformation component and a context retention mechanism to learn contextual information and combined contextual features with their transformed contextual features to obtain local salient features. Dong *et al.* [26] proposed a CNN with multiple non-linear transformations which has pursued a good result. Akhter *et al.* [27] achieved a great performance by introducing the single-layer CNN with multiple filters into document-level text classification.

At the same time, the attention-based model had also been proposed, which was used to capture the weight of each word within a sentence. Wang *et al.* [28] introduced the attention mechanism into LSTM, which provided a new idea for text classification. Chen *et al.* [29] introduced the product and user information of different semantic levels to classify texts through the attention mechanism. Liu and Zhang [30] proposed a method which introduces three attention mechanisms to determine the contribution of each word in the context in text classification. Gu *et al.* [31] proposed a position-aware bidirectional attention network based on the Bidirectional Gated Recurrent Unit(BiGRU) for text classification. Especially, Vaswani *et al.* [32] proposed the self-attention mechanism which is good at capturing the internal relevance between features and is less dependence on external information. The self-attention mechanism not only overcame the shortcoming of being unable to calculate parallelly on RNN, but also solved the problem of capturing the long-range dependency information difficultly on CNN. Dong *et al.* [33] obtained text representations containing more comprehensive semantics for text classification by introducing the Bidirectional Encoder Representation from Transformers(BERT) [34] and the self-interaction attention mechanism.

When processing text classification through the attention-based model, many breakthroughs in the field of node classification and edge prediction have been made with GCN in recent years. Hamilton *et al.* [35] proposed a variety of aggregation functions to learn the feature representation of each node to enhance the effect of GCN. Chen *et al.* [36] proposed a random training method. Their method can reduce the time complexity greatly with selecting two neighbor nodes for the convolution operation randomly. In the case of different sampling sizes, features will converge to a local optimum. Li *et al.* [37] proposed a method that can adaptively construct new Laplacian matrices based on tasks and generate different task-driven convolution kernels. This method is superior to GCN in processing multitask datasets. Velickovic *et al.* [38] used the attention mechanism to calculate the correlation between nodes dynamically and achieved good results in many public datasets. Yao *et al.* [39] introduced GCN for text classification and modeled the whole corpus into a heterogeneous network, which can learn word embeddings and document embeddings simultaneously. Cavallari *et al.* [40] introduced a new setting for graph embedding, which considers embedding communities instead of individual nodes. Although GCN performs well in text classification, it still fails to solve the problems of contextual dependency and lexical polysemy in text classification tasks. With addressing the two problems as a starting point, an improved model based on GCN is proposed accordingly.

III. IGCN

Through an in-depth research on text classification based on the neural network, IGCN is proposed in this paper, which builds three components of BiLSTM, the POS information, and the dependency relationship on GCN. The whole process of IGCN can be divided into steps of extracting the text feature, concatenating the POS feature, constructing the adjacency matrix based on the dependency relationship, training the neural network, and making a final prediction. The architecture of IGCN is shown in Fig. 2. Firstly, the text features and their corresponding POS features will be successively obtained by BiLSTM, which can utilize their respective contextual information effectively. Then, two kinds of features will be concatenated together to form the required feature of IGCN. Meanwhile, the dependency relationship will be generated to confront the problem of contextual dependency, and to construct the adjacency matrix required by IGCN. After that, the feature and the adjacency matrix will be input to train the neural network. With the hidden state vectors of the

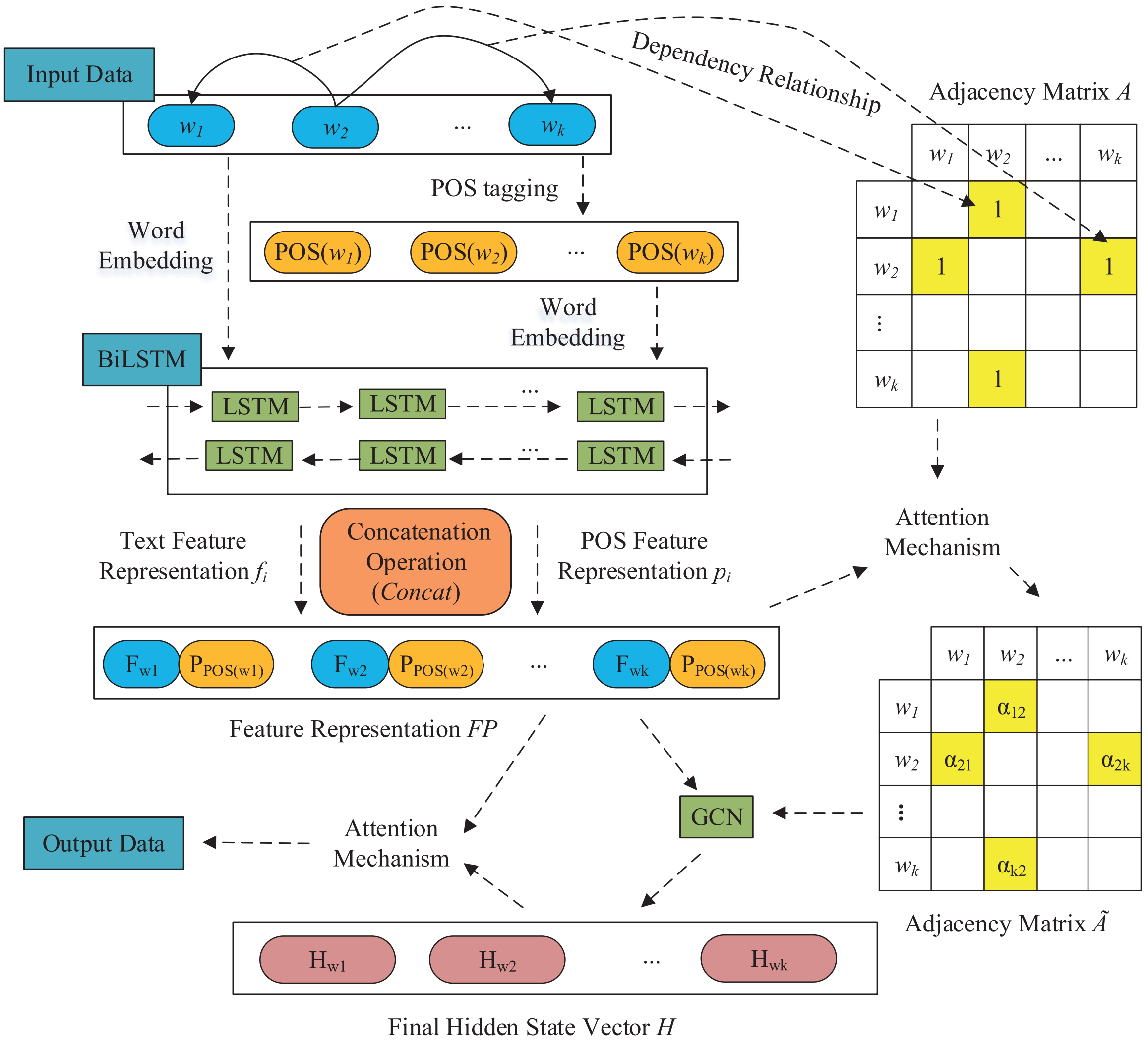


FIGURE 2. The architecture of IGCN.

last layer obtained, the weights between features and hidden state vectors can be calculated to determine the contribution of each word by the attention mechanism. Subsequently, the final features will be generated to predict the category of the given sentence.

A. THE BASICS OF IGCN: GCN

As known, the original GCN takes the pre-processed words as nodes and takes the relationship between words as edges. The original GCN can be divided into the input layer, the hidden layer, and the output layer. The architecture of the original GCN is shown in Fig. 3. A graph $G = (V, E)$ is given where V is the set of nodes and E is the set of edges between the nodes.

1) THE INPUT LAYER OF THE ORIGINAL GCN

The input layer of the original GCN consists of the input feature matrix and the adjacency matrix of the graph. The adjacency matrix is provided to express the reference relationship between the nodes. Let $X \in R^{N\times D}$ be the input feature matrix where N is the size of V, and D represents the dictionary set size of V. When one word in the i-th node is at the m-th position of the dictionary set, it can be expressed as $X_{im} = \{0_1, \ldots, 0_{m-1}, 1_m, 0_{m+1}, \ldots, 0_D\}$. Let $A \in R^{N\times N}$ be

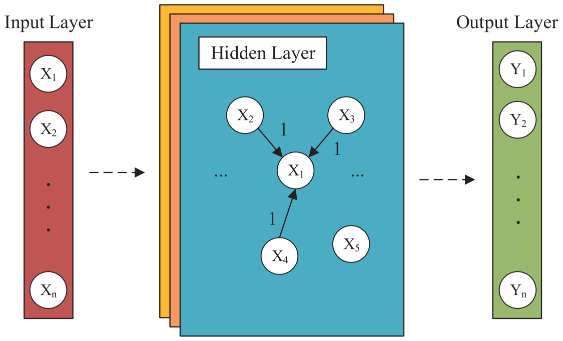


FIGURE 3. The architecture of the original GCN.

the self-loop adjacency matrix of the graph G. The adjacency matrix of an undirected graph can be expressed as follows.

$$A_{ij} = A_{ji} = \begin{cases} 0, i \nrightarrow j \\ 1, i \rightarrow j \\ 1, i = j \end{cases} \tag{1}$$

where A_{ij} represents that whether there is a connection between the i-th node and the j-th node.

2) THE HIDDEN LAYER OF THE ORIGINAL GCN

The hidden layer of the original GCN can aggregate the node information of the current layer through the propagation rules and transmit the features to the next layer. The features become more abstract as the propagation through successive hidden layers. The layer-wise propagation rules of the i-th node can be expressed as follows.

$$\begin{aligned} h_i^l &= \sigma(\sum_{j=1}^{N} \bar{A}_{ij} \cdot W^l \cdot h_i^{l-1} + b^l) \\ \bar{A} &= D^{-\frac{1}{2}} \cdot A \cdot D^{-\frac{1}{2}} \\ D_{ii} &= \sum_{j=1}^{N} A_{ij} \end{aligned} \tag{2}$$

where W^l is a trainable linear transformation weight which can be obtained by minimizing the cross-entropy loss function on all labeled samples. b^l is a bias term. $\bar{A} \in R^{N \times N}$ is the normalization adjacency matrix of the graph G. $D \in R^{N \times N}$ is the degree matrix of the graph G. σ denotes a nonlinear activation function (e.g. $ReLU$). h_i^l is the i-th node feature of the l-th hidden layer. Initially, $h_i^0 = X$.

3) THE OUTPUT LAYER OF THE ORIGINAL GCN

After obtaining the final features of the hidden layer, the output layer of the original GCN can generate the probability value of each category through the softmax function and classify the category of the text according to the maximum value of the probability.

B. THE INPUT LAYER OF IGCN

The quality of the initial feature directly affects the performance of the text classification model. In terms of feature extraction, the one-hot encoding features are generated with building a global text dictionary in the original GCN, which belong to the shallow feature. Therefore, to obtain more advanced and abstract features, BiLSTM is introduced to implement a deeper extraction of text features. On one hand, the text is a kind of the non-euclidean structure data. BiLSTM can retain the position information of text and capture the serialized features of text. On the other hand, the general social media text belongs to the typical short text. The gate mechanism of BiLSTM can effectively solve the problems of less contextual information and ambiguous semantics in such a short text. Besides, the bi-directional mechanism of BiLSTM ensures that each word can obtain more semantic information with full consideration of the context. Such a bi-directional model can provide a deeper text feature representation to the neural network. The architecture of BiLSTM is shown in Fig. 4.

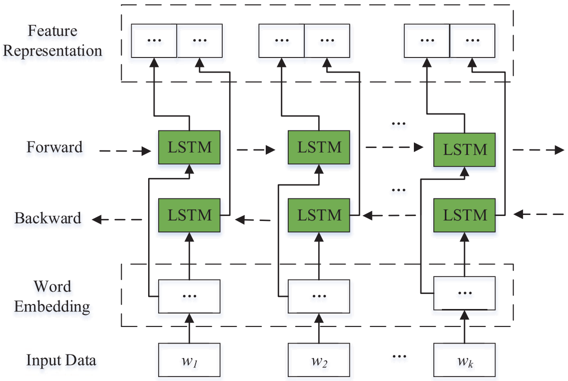


FIGURE 4. The architecture of BiLSTM.

Given a k-words sentence $S = \{W_1, W_2, \ldots, W_k\}$, the model embeds the the initial input text through the pre-trained embedding matrix. Then the text feature representation matrix $M \in R^{k \times d}$ can be obtained, where k is the vocabulary size of the sentence S and d represents the dimension of word embedding. Meanwhile, The text feature representations $F = \{f_1, f_2, \ldots, f_k\}$ of the sentence S with context information can also be got by BiLSTM, where $f_i \in R^{k \times df}$ represents the text feature of the i-th word. df is the dimension of the hidden state vector of BiLSTM.

In view of that the extraction of feature mentioned above is not sufficient to deal with the problem of lexical polysemy, the POS information of text is introduced to further eliminate the problem. Unlike the other attributes, the POS information is a basic grammatical attribute of the word which does not vary much with the difference of the field. As everyone knows, the short text lacks a certain degree of grammar and does not contain the whole rigorous structure information. Hence, more acceptable information will be provided by POS feature for short text classification.

Accordingly, the POS information will be input to BiLSTM to generate the POS feature representations $P = \{p_1, p_2, \ldots, p_k\}$, where $p_i \in R^{k \times df}$ is the POS feature of the i-th word.

Through the above feature extraction, the text feature representations F and the POS feature representations P have been obtained by BiLSTM. On this basis, the concatenation operation is chosen in this paper to effectively utilize the POS information to solve the problem of lexical polysemy. The text feature representations F and their corresponding POS feature representations P are concatenated by (3) to generate a deeper feature representations $FP = \{fp_1, fp_2, \ldots, fp_k\}$ which are also the feature representation required by IGCN. The concatenation operation is shown in Fig. 5.

$$fp_i = Concat(f_i, p_i) \tag{3}$$

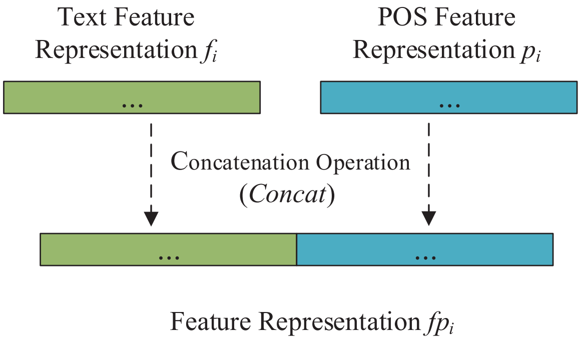

Text Feature Representation f_i
POS Feature Representation p_i
...
...
Concatenation Operation ($Concat$)
...
...
Feature Representation fp_i

FIGURE 5. The operation of concatenation.

where $fp_i \in R^{k\times 2df}$ represents the feature of the i-th word.

Given the loose syntactic constraint and the vague dependency, the dependency relationship is introduced to reveal a clear syntactic structure for text classification. The so-called dependency relationship is a more fine-grained attribute which can recognize the grammatical components in a sentence. To a certain extent, this paper also pays some attention to non-notional words (e.g. prepositions) in structure analysis. As shown in Fig. 6, the dependency relationship graph is applied to construct the adjacency matrix $A \in R^{k\times k}$ of a given sentence.

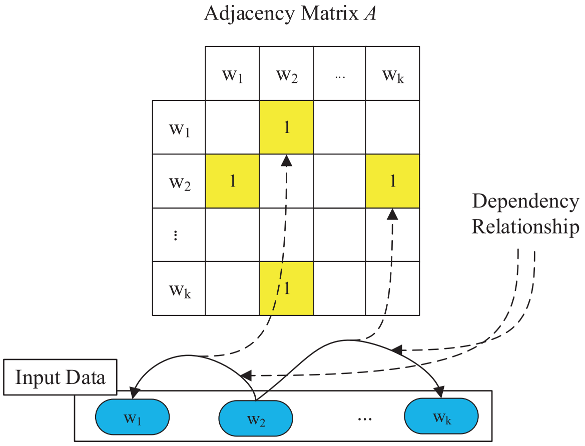


FIGURE 6. Adjacency matrix based on the dependency relationship.

C. THE HIDDEN LAYER OF IGCN

Compared with the adjacency matrix constructed by the dependencies of each node to its two neighbor nodes, the adjacency matrix based on the dependency relationship can provide the short-range contextual dependency and the long-range contextual dependency together, which is more effective to solve the problem of contextual dependency. Furthermore, an example is also used to explain how to construct the adjacency matrix with dependency relations in Fig. 7.

In the original GCN, the weights of edges are uniformly set to 1 so that the influences of each node to its neighbor nodes are same. However, not all edges have the same weight in reality and some of them must be important to text classification. For example, in the sentence "I like this movie so much.", the edge between the nodes "like" and "much" should contribute much more to text classification than the edge between the nodes "this" and "movie". Therefore, it is necessary to capture which neighbor nodes contribute more to the central node. To learn the contributions of neighbor nodes to the central node, the attention mechanism based on the dependency relationship is also introduced. The attention mechanism acts between the central node and neighbor nodes, which builds up the performance of the model with quantifying the contributions between nodes. As shown in Fig. 8, the adjacency matrix $A \in R^{k\times k}$ based on the attention mechanism is also obtained. In this paper, the similarity between nodes is selected to measure the dependency relationship. Based on the principle of not destroying the integrity of network, the topological information of the graph will be quantitatively analyzed from the perspective of local attributes and global attributes.

$$\begin{aligned}\tilde{A}_{ij} &= A_{ij}\cdot\alpha_{ij}\\ \alpha_{ij} &= soft\max(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{i=1}^{k}\exp(e_{ij})}\\ e_{ij} &= fp_i\cdot fp_j\end{aligned}\tag{4}$$

where (4) is used to calculate the influences between the i-th node and its neighbor nodes with the dependency relationship. The softmax function ensures that the sum of each row of the matrix is 1. In other words, the sum of the influences of all nodes in the sentence to the i-th node is 1 during the feature aggregation process. α_{ij} is the influence of the j-th node to the i-th node. e_{ij} is the similarity of the j-th node to the i-th node. A dot product operation is used to calculate the similarity between nodes. The larger the value of e_{ij} is, the more similar the i-th node and the j-th node are.

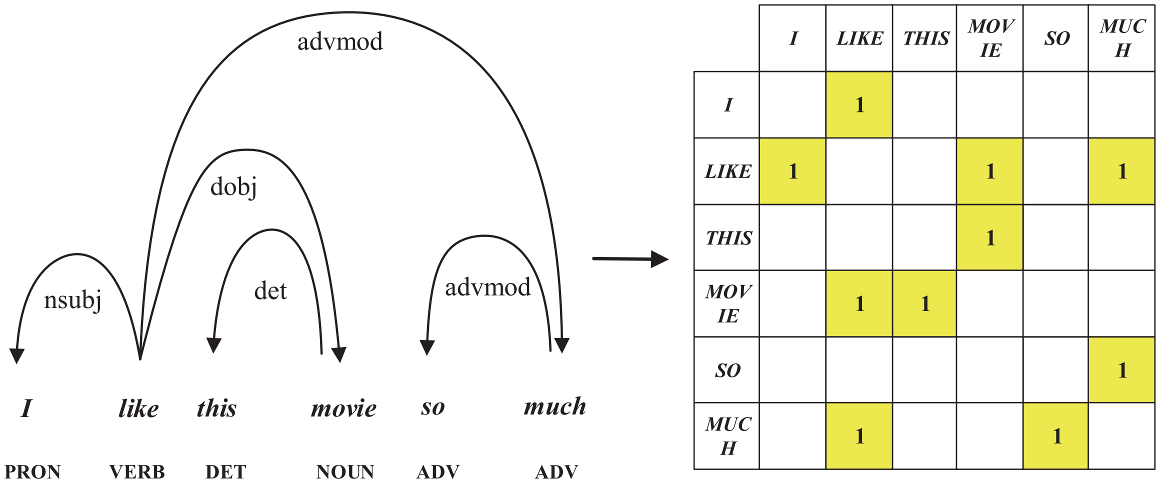
After the above data processing, the feature representations FP required by IGCN and the adjacency matrix $\tilde{A}$ based on the attention mechanism are both obtained. Subsequently, they will be fed into a single-layer GCN for training, and the hidden state vectors $H = \{h_1, h_2, \ldots, h_k\}$ can be obtained.

$$h_i = \sigma(\tilde{A}_i\cdot FP\cdot W_h + b_h)\tag{5}$$

where $h_i \in R^{k\times 2df}$ represents the hidden state vector of the i-th word, W_h is the trainable weight matrix, and b_h is a bias term.

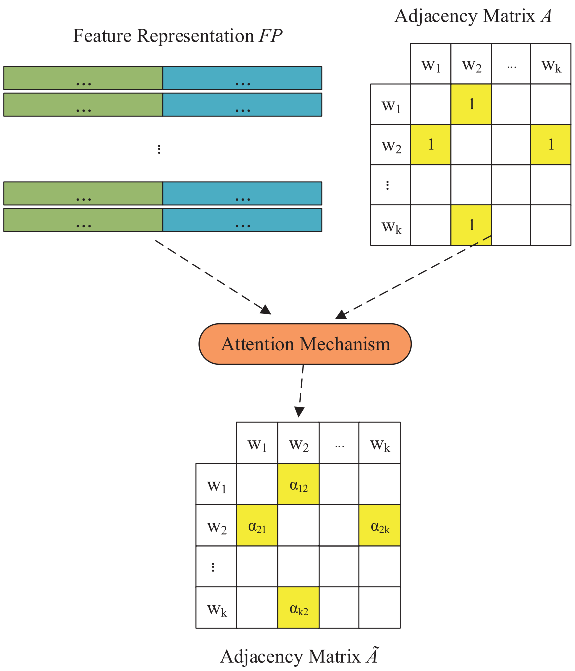
D. THE OUTPUT LAYER OF IGCN

With the feature representations FP and the hidden state vectors H obtained, the attention mechanism is introduced once more to find the important relevant semantical features. Unlike the general attention network, the contribution $\mu = \{\mu_1, \mu_2, \ldots, \mu_k\}$ of each word to text classification is obtained with linking the feature representations FP with



	I	LIKE	THIS	MOV IE	SO	MUC H
I		1				
LIKE	1			1		1
THIS				1		
MOV IE		1	1			
SO						1
MUC H		1			1	

FIGURE 7. An example to explain how to construct the adjacency matrix with dependency relations.



Adjacency Matrix *A*

	w_1	w_2	...	w_k
w_1		1		
w_2	1			1
⋮				
w_k		1		

	w_1	w_2	...	w_k
w_1		α_{12}		
w_2	α_{21}			α_{2k}
⋮				
w_k		α_{k2}		

Adjacency Matrix $\tilde{A}$

FIGURE 8. Adjacency matrix based on the attention mechanism.

the hidden state vectors H in the output layer. Meanwhile, the final feature representation y of the sentence is also obtained as follows.

$$\begin{aligned} y &= \mu \cdot FP \\ \mu_i &= soft\max(\lambda_i) = \frac{\exp(\lambda_i)}{\sum\limits_{i=1}^{k} \exp(\lambda_i)} \\ \lambda_i &= \sum_{j=1}^{k} h_i \cdot fp_j^T \end{aligned} \tag{6}$$

where μ_i is the contribution of the i-th node to text classification and λ_i is the sum of the similarity between the i-th node and all nodes in the sentence.

Subsequently, the final feature representation y is input into the fully-connected layer and the category of text will be predicted by the softmax function.

$$Y = soft\max(y \cdot W + b) \tag{7}$$

where W is a trainable weight matrix and b is a bias term.

IV. EXPERIMENTS

In this section, this paper first describes how to set up the experiment and analyzes the experimental results of different models. Subsequently, the ablation study is implemented to further prove the performance brought by each component.

A. DATASETS AND PARAMETER SETTINGS

As shown in Table 1, the following three datasets are adopted in the experiment.

- IMDB (Internet Movie DataBase) dataset, which belongs to the binary-classification English dataset, contains 10,662 pieces of the internet movie review data provided by the data competition platform Kaggle. The training set contains 6,000 pieces of data and the test set contains 4,662 pieces of data, where the positive comments and the negative comments in the training set and the test set account for 50% respectively.
- AAR (Amazon Alexa Reviews) dataset, which belongs to the multiclass-classification English dataset, contains 3,150 pieces of the Amazon Alexa product review data collected in 2018. The training set contains 1,575 pieces of data, and the test set contains 1,575 pieces of data.
- TUA (Twitter US Airline) dataset, which belongs to the multiclass-classification English dataset, contains 14,640 pieces of Twitter user comments on American Airlines collected in 2015. The training set contains

TABLE 1. Statistics of dataset.

DataSet	Training Set	Test Set	Category
IMDB	6000	4662	2
AAR	1575	1575	5
TUA	7320	7320	3

7,320 pieces of data and the test set contains 7,320 pieces of data. The negative comments, the neutral comments, and the positive comments in the training set and the test set account for 63%, 21%, and 16% respectively.

Two pre-trained vector matrices of Yelp and Sogou News are used in this paper to achieve the initial word embedding. The initialization of weight matrix randomly adopts three methods: uniformly distributed initialization, normal distribution initialization, and orthogonal initialization. All the significant parameters in the experiment are recorded in Table 2.

TABLE 2. Experimental parameter settings.

Experimental parameter	Value
Dropout rate	0.3
Number of GCN layer	1
Dimension of word embedding	300
Maximum number of iterations	100
Learning rate	10^{-3}
Weight decay of L2 regularization	10^{-5}
Batch size	32
Dimension of the hidden state vector	300

In the training process, the dropout is used to introduce the randomness and prevent overfitting in this paper, and the nonlinear function *LeakyReLU* is selected as the activation function of GCN to overcome the problem of vanishing gradient and to speed up the training speed. The cross-entropy loss function is used to train the model.

$$LeakyReLU(x) = \begin{cases} 0.2x, x < 0 \\ x, x \geq 0. \end{cases} \tag{8}$$

Besides, the L_2 regularization is also used to reduce the overfitting. The adaptive moment estimation optimizer is selected as the model optimizer. The stopping condition of the training process is that the loss function has not decreased for 10 consecutive iteration cycles.

To comprehensively evaluate the model, the accuracy is selected as the evaluation metric and IGCN is compared with the following baseline models.

- SVM is a binary-classification model of the supervised learning. It is commonly applied for text classification, pattern recognition, and regression analysis. Moreover, SVM is mainly oriented to small samples, nonlinear data, and high-dimensional data.
- TextCNN is a specific form of CNN. It obtains the shallow feature representation of the sentence effectively with the one-dimensional convolution operation. Moreover, it performs well in the field of short text classification, such as the search field and the dialogue field.
- LSTM is a specific form of RNN, which achieves good results in both the short text classification and the long text classification.
- BiLSTM is a specific form of LSTM, which further captures bidirectional semantic dependencies.
- GCN is a classification model based on the non-euclidean structure data, which is gradually emerging in the field of text classification.
- GAT is a classification model which borrows the attention mechanism into GCN.
- TextGCN is a text classification model with utilizing the document-word relation and the word-word relation, which gains a variety of improvements over many state-of-the-art models.

B. MAIN EXPERIMENTAL RESULTS

As shown in Table 3, the experimental results on three benchmarking datasets confirm that the performance of IGCN is better than other baseline models, which further prove the effectiveness and robustness of IGCN in text classification.

TABLE 3. Experimental results of different classification models.

Model	IMDB	AAR	TUA
SVM	0.7775	0.6466	0.6940
TextCNN	0.7630	**0.7115**	0.7105
LSTM	0.7555	0.6984	0.6956
BiLSTM	0.7595	0.6940	0.7008
GCN	0.7390	0.6358	0.6836
GAT	0.7589	0.6514	0.6995
TextGCN	0.7597	0.6496	0.6911
IGCN	**0.8085**	0.6807	**0.7382**

As shown in Fig. 9, the accuracy of SVM on the IMDB dataset reaches 77.75%, which is higher than the accuracies of the six models based on neural networks (TextCNN, LSTM, BiLSTM, GCN, GAT, and TextGCN). On the AAR dataset and the TUA dataset, the accuracies of SVM are lower than the accuracies of the other four models based on the neural network(TextCNN, LSTM, BiLSTM, and GAT), which only reach 64.66% and 69.4%. Accidentally, the accuracy of SVM is higher than the accuracy of GCN. It shows that SVM is weak to tackle the large-scale training samples and the multi-classification problems. Besides, the accuracies of TextCNN on the three benchmarking datasets are surprisingly

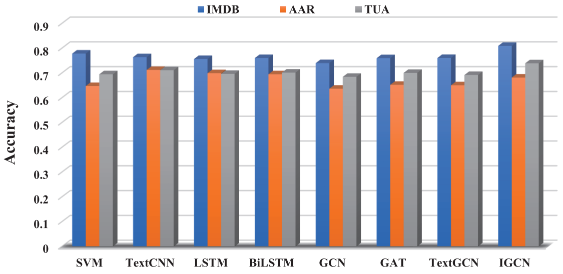


FIGURE 9. Histogram of experimental results of different classification models.

higher than the accuracies of LSTM, BiLSTM, GCN, GAT, and TextGCN. It means that the text classification model may pay more attention to the short-range dependency information on the three benchmarking datasets.

As shown in Fig. 10 and Fig. 11, the loss value of IGCN on the IMDB dataset decreases faster than the loss value of GCN and the accuracy of IGCN on the IMDB dataset increases faster than the accuracy of GCN. The results show that IGCN can improve the effectiveness and achieve better results in less iteration.

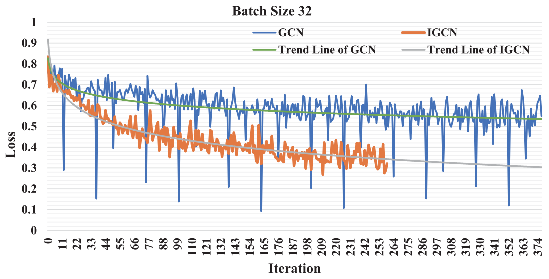


FIGURE 10. The loss curve of GCN and IGCN.

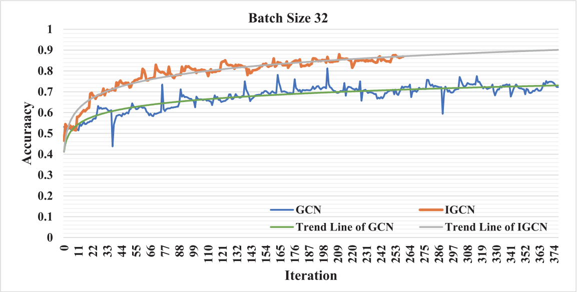


FIGURE 11. The accuracy curve of GCN and IGCN.

Compared with the other models based on neural network, the accuracies of the graph-based GCN are lowest on the three benchmarking datasets, which reach 73.9%, 63.58%, and 68.36% respectively. The reason is that GCN cannot make full use of contextual dependency information in the short text classification due to the sparse adjacency matrix constructed by GCN. At the same time, compared with GCN, IGCN boosts the performance on the IMDB, AAR, and TUA datasets. As shown in Table 3, IGCN increases the accuracies by 6.95%, 4.49%, and 5.46% respectively, which fully proves the effectiveness of IGCN. There are two reasons for this result.

- The multiple effects of the pre-trained word embedding, the BiLSTM, and the POS information are fully utilized in this paper to extract more advanced and abstract feature representations to further solve the problem of lexical polysemy. At the same time, the dependency relationship introduced in this paper can make reasonable use of the short-range contextual dependency and the long-range contextual dependency together.
- The attention mechanism introduced in this paper captures not only the importances of different neighbor nodes, but also the importances of different types of nodes. It can reduce the weight of noise nodes to a certain extent.

C. ADDITIONAL EXPERIMENTAL RESULTS

Whether using the same BiLSTM to generate the different kinds of feature representations is studied on the IMDB dataset. To analyze its impact on the performance of IGCN, there are three cases listed as follows.

- One case is that the same BiLSTM is used to train the text features F and the POS feature P successively, which is used in IGCN.
- Another case is that the same BiLSTM is used to train the POS feature P and text features F successively, which is named as MPF.
- The last case is that two BiLSTMs are used to train the feature, which is named as M2Bi.

As shown in Table 4, the experimental results on the three datasets illustrate that the accuracies of IGCN have increased by 0.9%, 1.02%, and 1.33% respectively compared with M2Bi. Fig. 12 proves that using the same BiLSTM to generate the text feature and the POS feature can improve the overall performance of IGCN.

TABLE 4. Experimental results of whether to use the same BiLSTM.

Model	IMDB	AAR	TUA
IGCN	**0.8085**	**0.6807**	**0.7382**
MPF	0.7725	0.6587	0.7016
M2Bi	0.7995	0.6705	0.7249

Moreover, the effect of the number of layers on the performance of IGCN is also studied on the IMDB dataset. To analyze the impact, the number of layers would be set as shown in Fig. 13. The results show that IGCN achieves the best performance when the number of layer is 1. Meanwhile, it is found that the accuracy decreases when the number of layer increases.

D. ABLATION STUDY

To further examine the benefit brought by each component of IGCN, an ablation study is implemented in this paper. The results of IGCN on the three datasets are presented as the

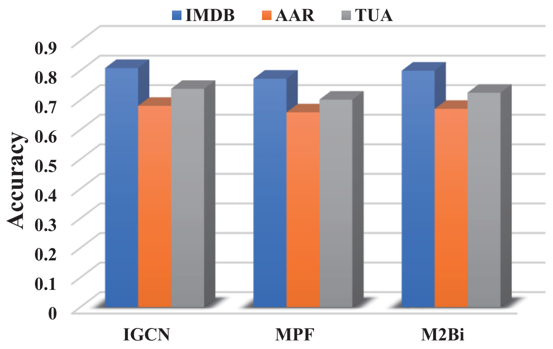


FIGURE 12. Histogram of experimental results of whether to use the same BiLSTM.

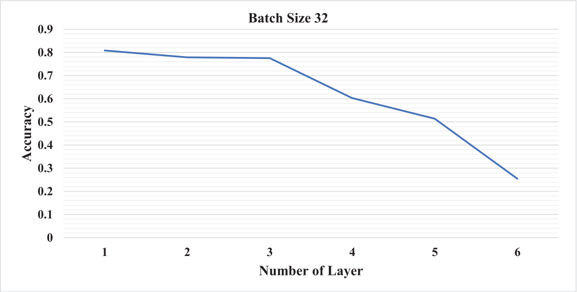


FIGURE 13. The accuracy curve of different layer number.

baseline. As shown in Table 5, the experimental results of IGCN and the three variant models prove that the three components applied in this paper can promote the performance of IGCN.

TABLE 5. Experimental results of ablation study.

Model	IMDB	AAR	TUA
IGCN	**0.8085**	**0.6807**	**0.7382**
MRD	0.6268	0.5244	0.5617
MRP	0.7555	0.6178	0.6902
MRBi	0.6472	0.5415	0.5767

After the dependency relationship is removed, the model named as MRD is only constructed by learning the dependency information between each node and its two neighbor nodes. As shown in Fig. 14, the accuracies of MRD on the IMDB, AAR, and TUA datasets decrease by 18.17%, 15.63%, and 17.65% respectively compared with the accuracies of IGCN. The average decrease rate of its accuracy reaches 17.15%. It shows that the dependency relationship is significantly relevant to text classification based on the graph. The weights of neighbor nodes to the central node can be deeply learned to further boost the performance of model by the attention mechanism on the dependency relationship.

Subsequently, after the POS feature is removed, the performance of the model named as MRP also drops. As shown in Fig. 14, the accuracies of MRP on the three datasets are lower than the accuracies of IGCN, which decrease by 5.3%, 6.29%, and 4.8% respectively. The experimental results demonstrate that the introduction of the POS information can solve the problem of lexical polysemy. With the average decrease rate of the accuracy reaching 5.46%, it means that the POS information works in text classification tasks.

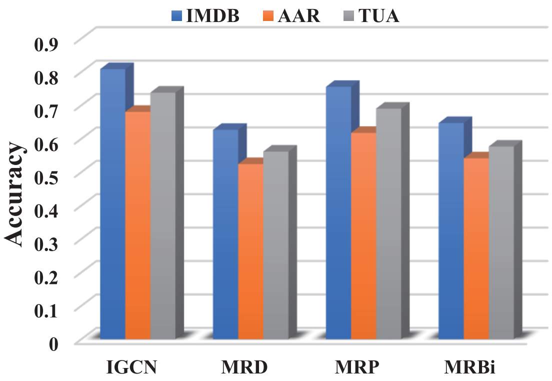


FIGURE 14. Histogram of experimental results of ablation study.

Finally, after BiLSTM is removed, the performance of the model named as MRBi also drops. As shown in Fig. 14, compared with the accuracies of IGCN, the accuracies of MRBi on the IMDB, AAR, and TUA datasets decrease by 16.13%, 13.92%, and 16.17% respectively. The average decrease rate of the accuracy reaches 15.41%. Its decrease of the accuracy proves that BiLSTM can enhance the performance of IGCN with the contextual dependency information.

To sum up, the decreases of the accuracies of MRD, MRP, and MRBi prove that each component of IGCN proposed in this paper is beneficial to address the problems in text classification.

V. CONCLUSION AND FUTURE WORK

Based on a review of the existing challenges faced in text classification, the applicability of the GCN-based model is discussed. In this paper, an improved integration model based on GCN is formulated, analyzed, and performed to handle the problems of contextual dependency and lexical polysemy. Experiments on three datasets show that IGCN performs better than other models, especially than GCN. Thus, the success of IGCN can be attributed to the integration of three components. On the one hand, the BiLSTM and the POS information can generate deeper feature representations for text classification. On the other hand, the dependency relationship provides a required graph fo GCN which can be used to capture the short-range contextual dependency and the long-range contextual dependency together. Due to the good performance with the reasonable combination of dependency relationship and GCN, it will certainly lead to a further exploration of the dependency relationship in the future. What is more, the transplantation of the dependency relationship from relation extraction tasks to text classification tasks can also give a new idea to process the non-euclidean structure data.

There is also a domanial problem in text classification. Some text in different fields has the same literal meaning. However, its importance is different for text classification. For example, the word ‘‘cancer’’ is a neutral tendency in the medical field but a negative tendency in other fields. The difference of the tendency may cause a wrong prediction. Therefore, an in-depth study will be carried out in the domain knowledge transfer.

REFERENCES

[1] W. Zhao, G. Zhang, G. Yuan, J. Liu, H. Shan, and S. Zhang, ''The study on the text classification for financial news based on partial information,'' *IEEE Access*, vol. 8, pp. 100426–100437, 2020.

[2] K. Liu and L. Chen, ''Medical social media text classification integrating consumer health terminology,'' *IEEE Access*, vol. 7, pp. 78185–78193, 2019.

[3] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, ''Deep learning based text classification: A comprehensive review,'' 2020, *arXiv:2004.03705*. [Online]. Available: http://arxiv.org/abs/2004.03705

[4] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu, ''A survey on knowledge graphs: Representation, acquisition and applications,'' 2020, *arXiv:2002.00388*. [Online]. Available: http://arxiv.org/abs/2002.00388

[5] C. Cortes and V. Vapnik, ''Support-vector networks,'' *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[6] Y. Kim, ''Convolutional neural networks for sentence classification,'' in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1746–1751.

[7] H. Zhang, L. Xiao, Y. Wang, and Y. Jin, ''A generalized recurrent neural architecture for text classification with multi-task learning,'' in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 2873–2879.

[8] W. Zhao, H. Peng, S. Eger, E. Cambria, and M. Yang, ''Towards scalable and reliable capsule networks for challenging NLP applications,'' in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 1549–1559.

[9] M. Schuster and K. K. Paliwal, ''Bidirectional recurrent neural networks,'' *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.

[10] S. Hochreiter and J. Schmidhuber, ''Long short-term memory,'' *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[11] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, ''Learning phrase representations using RNN encoder–Decoder for statistical machine translation,'' in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1724–1734.

[12] R. Wang, Z. Li, J. Cao, T. Chen, and L. Wang, ''Recurrent convolutional neural networks for text classification,'' in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 2267–2273.

[13] R. Wang, Z. Li, J. Cao, T. Chen, and L. Wang, ''Convolutional recurrent neural networks for text classification,'' in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–6.

[14] T. N. Kipf and M. Welling, ''Semi-supervised classification with graph convolutional networks,'' in *Proc. 5th Int. Conf. Learn. Represent. (ICLR)*, Apr. 2017, pp. 1–14.

[15] G. Li, M. Muller, A. Thabet, and B. Ghanem, ''DeepGCNs: Can GCNs go as deep as CNNs?'' in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9266–9275.

[16] J. Liu, F. Meng, Y. Zhou, and B. Liu, ''Character-level neural networks for short text classification,'' in *Proc. Int. Smart Cities Conf. (ISC2)*, Sep. 2017, pp. 560–567.

[17] X. Zhang, J. J. Zhao, and Y. LeCun, ''Character-level convolutional networks for text classification,'' in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2015, pp. 649–657.

[18] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao, and B. Xu, ''Attention-based bidirectional long short-term memory networks for relation classification,'' in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics (Short Papers)*, vol. 2, 2016, pp. 207–212.

[19] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, ''Deep contextualized word representations,'' in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol., (Long Papers)*, vol. 1, 2018, pp. 2227–2237.

[20] D. Tang, B. Qin, X. Feng, and T. Liu, ''Effective LSTMs for target-dependent sentiment classification,'' in *Proc. 26th Int. Conf. Comput. Linguistics (COLING)*, Dec. 2016, pp. 3298–3307.

[21] M. Zhang, Y. Zhang, and D. Vo, ''Gated neural networks for targeted sentiment analysis,'' in *Proc. 30th AAAI Conf.*, Feb. 2016, pp. 3087–3093.

[22] M. Yang, Q. Qu, X. Chen, C. Guo, Y. Shen, and K. Lei, ''Feature-enhanced attention network for target-dependent sentiment classification,'' *Neurocomputing*, vol. 307, pp. 91–97, Sep. 2018.

[23] W. Xue and T. Li, ''Aspect based sentiment analysis with gated convolutional networks,'' in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics (Long Papers)*, vol. 1, 2018, pp. 2514–2523.

[24] B. Huang and K. Carley, ''Parameterized convolutional neural networks for aspect level sentiment classification,'' in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 1091–1096.

[25] X. Li, L. Bing, W. Lam, and B. Shi, ''Transformation networks for target-oriented sentiment classification,'' in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics (Long Papers)*, vol. 1, 2018, pp. 946–956.

[26] M. Dong, Y. Li, X. Tang, J. Xu, S. Bi, and Y. Cai, ''Variable convolution and pooling convolutional neural network for text sentiment classification,'' *IEEE Access*, vol. 8, pp. 16174–16186, 2020.

[27] M. P. Akhter, Z. Jiangbin, I. R. Naqvi, M. Abdelmajeed, A. Mehmood, and M. T. Sadiq, ''Document-level text classification using single-layer multisize filters convolutional neural network,'' *IEEE Access*, vol. 8, pp. 42689–42707, 2020.

[28] Y. Wang, M. Huang, X. Zhu, and L. Zhao, ''Attention-based LSTM for aspect-level sentiment classification,'' in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 606–615.

[29] H. Chen, M. Sun, C. Tu, Y. Lin, and Z. Liu, ''Neural sentiment classification with user and product attention,'' in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 1650–1659.

[30] J. Liu and Y. Zhang, ''Attention modeling for targeted sentiment,'' in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics, Short Papers*, vol. 2, Apr. 2017, pp. 572–577.

[31] S. Gu, L. Zhang, Y. Hou, and Y. Song, ''A position-aware bidirectional attention network for aspect-level sentiment analysis,'' in *Proc. 27th Int. Conf. Comput. Linguistics (COLING)*, Aug. 2018, pp. 774–784.

[32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, ''Attention is all you need,'' in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2017, pp. 5998–6008.

[33] Y. Dong, P. Liu, Z. Zhu, Q. Wang, and Q. Zhang, ''A fusion model-based label embedding and self-interaction attention for text classification,'' *IEEE Access*, vol. 8, pp. 30548–30559, 2020.

[34] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, ''BERT: Pre-training of deep bidirectional transformers for language understanding,'' in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1, Jun. 2019, pp. 4171–4186.

[35] W. L. Hamilton, Z. Ying, and J. Leskovec, ''Inductive representation learning on large graphs,'' in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2017, pp. 1024–1034.

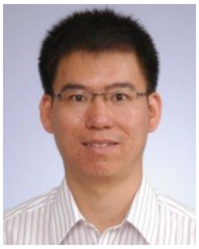
[36] J. Chen, J. Zhu, and L. Song, ''Stochastic training of graph convolutional networks with variance reduction,'' in *Proc. 35th Int. Conf. Mach. Learn. (ICML)*, Jul. 2018, pp. 941–949.

[37] R. Li, S. Wang, F. Zhu, and J. Huang, ''Adaptive graph convolutional neural networks,'' in *Proc. 32th AAAI Conf.*, Feb. 2018, pp. 3546–3553.

[38] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, ''Graph attention networks,'' in *Proc. 6th Int. Conf. Learn. Represent. (ICLR)*, Apr. 2018, pp. 1–12.

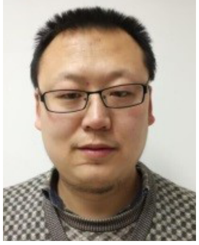
[39] L. Yao, C. Mao, and Y. Luo, ''Graph convolutional networks for text classification,'' in *Proc. 31th AAAI Conf.*, Jan. 2019, pp. 7370–7377.

[40] S. Cavallari, E. Cambria, H. Cai, K. C.-C. Chang, and V. W. Zheng, ''Embedding both finite and infinite communities on graphs [application notes],'' *IEEE Comput. Intell. Mag.*, vol. 14, no. 3, pp. 39–50, Aug. 2019.



HENGLIANG TANG received the B.Sc. and Ph.D. degrees from the Beijing University of Technology, in 2005 and 2011, respectively. He is currently a Professor with Beijing Wuzi University. His main research interests include computer vision and the IoT information technology.

YUAN MI received the B.Sc. degree from Northwest A&F University, in 2015. He is currently pursuing the M.Sc. degree with Beijing Wuzi University. His main research interests include natural language processing and the IoT information technology.



FEI XUE received the B.Sc. degree from the University of Jinan, in 2006, the M.Sc. degree from the Taiyuan University of Science and Technology, in 2011, and the Ph.D. degree from the Beijing University of Technology, in 2016. He is currently an Associate Professor with Beijing Wuzi University. His main research interests include computational complexity theory and optimization.

YANG CAO received the B.Sc. and M.Sc. degrees from the Taiyuan University of Science and Technology, in 2011 and 2015, respectively, and the Ph.D. degree from the Beijing University of Technology, in 2019. He is currently a Lecturer with Beijing Wuzi University. His main research interests include machine learning and big data analysis.