

# Public Sentiment Analysis of Social Security Emergencies Based on Feature Fusion Model of BERT and TextLevelGCN

Linli Wang, Hu Wang, Hanlu Lei

School of Management, Wuhan University of Technology, Wuhan, China

Email: fleihanlu@163.com

**How to cite this paper:** Wang, L.L., Wang, H. and Lei, H.L. (2023) Public Sentiment Analysis of Social Security Emergencies Based on Feature Fusion Model of BERT and TextLevelGCN. *Journal of Computer and Communications*, 11, 194-204.  
<https://doi.org/10.4236/jcc.2023.115014>

**Received:** April 13, 2023

**Accepted:** May 28, 2023

**Published:** May 31, 2023

---

## Abstract

At present, the emotion classification method of Weibo public opinions based on graph neural network cannot solve the polysemy problem well, and the scale of global graph with fixed weight is too large. This paper proposes a feature fusion network model Bert-TextLevelGCN based on BERT pre-training and improved TextGCN. On the one hand, Bert is introduced to obtain the initial vector input of graph neural network containing rich semantic features. On the other hand, the global graph connection window of traditional TextGCN is reduced to the text level, and the message propagation mechanism of global sharing is applied. Finally, the output vector of BERT and TextLevelGCN is fused by interpolation update method, and a more robust mapping of positive and negative sentiment classification of public opinion text of “Tangshan Barbecue Restaurant beating people” is obtained. In the context of the national anti-gang campaign, it is of great significance to accurately and efficiently analyze the emotional characteristics of public opinion in sudden social violence events with bad social impact, which is of great significance to improve the government’s public opinion warning and response ability to public opinion in sudden social security events.

## Keywords

Social Security Emergencies, Network Public Opinion Emotion Analysis, Graph Neural Network, TextLevelGCN, BERT

---

## 1. Introduction

According to the Emergency Response Law promulgated [1] in 2007, emergencies include public health events, natural disasters, accident disasters and social

security events. Among them, social security emergencies are defined as “mass incidents caused by internal contradictions among the people, or accumulated and stimulated by improper handling of internal contradictions among the people, involving part of the public, with certain organizations and destinations taking various types of behaviors, affecting government management and social order, and even causing the society to fall into a certain intensity of confrontation within a certain range.” Compared with other emergencies, social security emergencies are more likely to cause huge waves in cyberspace and become the focus and hot spot of citizens’ attention. As such incidents are more subjective and social [2], they are often caused by people’s subjective intentions. Improper handling of public opinion may have an indirect impact on the investigation and judgment of cases. Therefore, paying attention to the emotional tendency and evolution process of netizens in the public opinion of social security emergencies has guiding significance for the government to effectively guide, prevent and manage the public opinion of social security in different stages of the development of the event.

Aiming at the social security emergency “Beating someone in Tangshan Barbecue restaurant”, this paper constructs the Bert-TextLevelGCN deep neural network model to obtain a robust prediction of Weibo users’ emotional polarity. BERT is the upstream part and TextLevelGCN is the downstream part. BERT can learn the statistical features of neighboring words, while TextLevelGCN can extract more local features and learn more semantic features according to the text context, which accords with the logic of human language system. Moreover, BERT’s strong word embedding dimension reduction ability, combined with the separate text graph constructed by TextLevelGCN and the reduced moving window, can effectively reduce the size of text graph, so as to save running memory and improve the efficiency of text classification tasks. Combined with the empirical analysis results, it has important reference value to put forward effective supervision and guidance for major public opinion events caused by social security emergencies.

## 2. Related Research

For the sentiment analysis of online public opinions, the early studies were mostly based on the sentiment dictionary matching method or traditional machine learning algorithm. The analysis methods based on emotion dictionaries mainly use emotion words for emotion analysis. For example, Aggarwal *et al.* put forward several methods of constructing emotion dictionaries to mark and match emotion polarity, thus achieving the scoring and classification of emotion types [3]. However, currently, the most widely studied method in the field of text sentiment analysis is the method based on deep learning [4]. Deep learning constructs a network model by simulating the human brain nervous system to learn the text, and automatically extracts features from the original data without manual design. Faced with the processing of massive data, Compared with machine

learning in modeling, migration, optimization and other aspects, the advantages of CNN are more obvious. Chen [5] first proposed to use CNN in text sentiment analysis, but CNN can only extract local features and cannot capture long-distance dependence.

For the mixed data of many unstructured data and text, image, speech and other data, powerful as convolutional Neural Network, it is difficult to directly process. Therefore, a special Graph Neural Network (GNN) based on deep learning was born. Peng *et al.* put forward a GNN-based model to convert text into text graph, and use GNN to learn local features at different levels and text word graph to obtain the advantages of discontinuous semantics and long-distance semantics. Meanwhile, the interdependence between classification labels is constructed. Compared with the hierarchical classification method based on traditional deep learning, the effect on large-scale hierarchical classification data sets has been significantly improved [6]. GNN model can be subdivided into Graph Convolution Networks [7], Graph Attention Networks [8], Graph Auto encoders [9], Graph Generative Networks [10] and Graph Spatial-temporal Networks [11]. Yao *et al.* [12], on the basis of not changing the network structure of GCN model, built a global text graph for the whole corpus based on the relationship between documents and words and the co-occurrence of words. Double-layer GCN is used to continuously learn new representations in the training process layer, and then the obtained text representations are input into the full-connection layer to obtain the classification probability, achieving better results in multiple text classification tasks. In order to solve the problem that text-GCN cannot be tested online and the expression of graph network is limited.

The pre-training model emerged in 2018. The word vector generated by ELMO, a new language model constructed by Peters *et al.* [13], can dynamically transform polysemical words with the context. The more powerful BERT model is composed of pre-training and fine-adjustment. Due to the limitations of the single model, many scholars combine the pre-training model and different deep learning models. To capture deeper semantic information, more and more scholars combine it with graph neural network model and achieve better classification effect. Wu [14] fused BERT and GCN to extract rich semantic information from large-scale corpora by using BERT. In the processing of GCN, a spatial updating method—message propagation mechanism based on word weight was adopted to obtain some local graph information of the contained text graph, and finally the representation of the whole text graph was obtained. It is also used to predict text emotional labels.

### 3. Construction of Bert-TextLevelGCN Model

The Bert-TextLevelGCN model consists of three parts: the first part is BERT pre-training, which obtains the word embedding of the initial word node containing rich semantic feature information. The second part is TextLevelGCN. Firstly, input the initializing word embedding matrix formed by BERT, construct

word level text graph for each text, and then obtain word embedding with precise meaning in specific context through the message propagation mechanism connecting context adjacency nodes. The third part is classification prediction: the emotion classification probability of TextLevelGCN module and the classification probability of BERT module acting alone are obtained by full connection layer and activation function respectively. The fourth part is feature fusion: the two kinds of classification probabilities in the third part are weighted together by interpolation updating method to get the fusion classification probability.

In order to enable GCN to better learn the semantic feature expression of text, this paper shrinks the sliding window and constructs text graph for each text when fusing Bert and TextLevelGCN models. In order to solve the problem of gradient disappearance or excessive smoothing when gradient return is encountered, the interpolation updating method is used in this paper. The weighted sum of the prediction classification probability  $Z_{\text{TextLevelGCN}}$  obtained by TextLevelGCN model and the prediction classification probability  $Z_{\text{BERT}}$  obtained by Bert acting alone on the text is used to obtain the fusion classification probability  $Z$ . Then, the cross entropy loss function is minimized to achieve the final emotion classification prediction, which can be expressed as:

$$Z = \lambda Z_{\text{TextLevelGCN}} + (1 - \lambda) Z_{\text{BERT}}. \quad (1)$$

In the above equation, when  $\lambda = 1$ , the BERT part is not updated; When  $\lambda = 0$ , the TextLevelGCN section is not updated; The update of the two models is controlled by adjusting the values between  $\lambda \in (0,1)$ . The structure diagram of Bert-TextLevelGCN model is shown in **Figure 1**.

### 3.1. Bert Pre-Training Model

BERT is a pre-trained language model proposed by Google's Devlin *et al.* [15]. It is derived from the Transformer model, which has an Encoder-Decoder structure. BERT only uses its Encoder part to obtain the feature representation of the text through multiple layers of overlapping. In the BERT word embedding structure, enter a text vector consisting of  $n$  characters, represented as  $\{w_1, w_2, \dots, w_n\}$ . After being trained by multiple layers of bidirectional Transformer encoders, it outputs a vector that contains the semantic information of the text. The initialized word embedding is represented as  $\{r_1, r_2, \dots, r_n\}$ .

BERT's training uses the Masked Language Model (MLM) and Next Sentence Prediction (NSP). MLM randomly masks a certain proportion of characters, and the model learns the masked words or characters through global context to predict the masked or replaced words, thereby achieving bidirectional encoding. NSP can be seen as a sentence-level binary classification problem. The sentence begins with [CLS], and [SEP] is added at the end of a sentence between two sentences. The logical relationship between sentences is mined by judging whether the next sentence is a reasonable next sentence for the previous sentence. By combining these two unsupervised learning tasks, more complete word embeddings are obtained, and the characterization of the semantic information of the input text is more accurate.

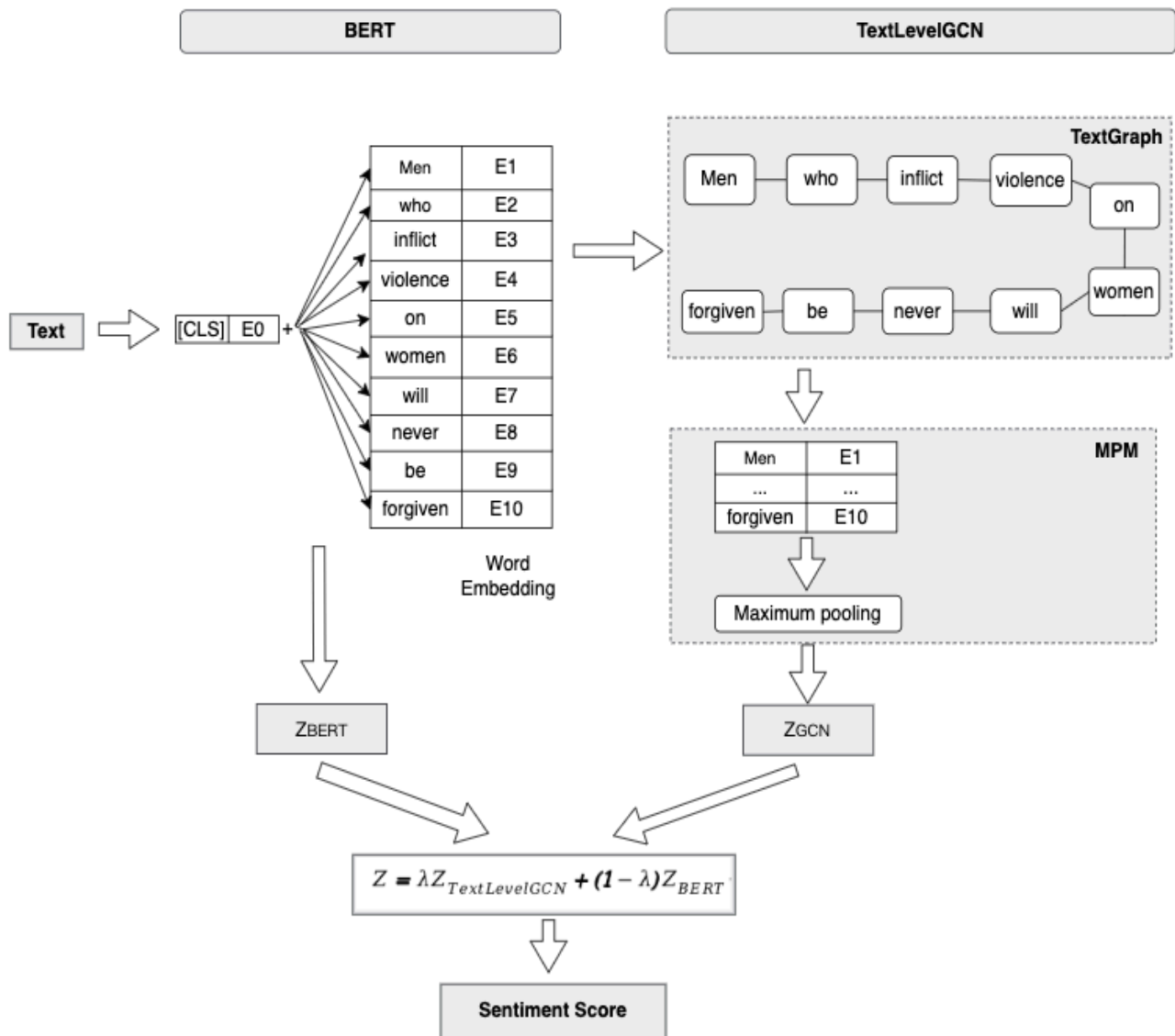


Figure 1. BERT-TextLevelGCN model structure diagram.

### 3.2. TextLevelGCN Model

When processing text classification tasks, BERT model is used to supplement text semantic features for TextGCN, thus achieving a good effect, but there are still two problems: 1) BertGCN constructs a heterogeneous graph for the whole corpus, Such fixed weight limits the expression ability of edges. In addition, in order to obtain a global representation, we have to use a very large connection window, which makes the constructed graph very large, there are many edges, and the model needs to consume a lot of memory. 2) It is impossible to classify the new text, because the structure and parameters of the graph depend on the corpus and cannot be modified again after the training.

To solve the above two problems, this paper uses TextLevelGCN model to improve the original GCN model: 1) A separate graph is constructed for each input text, with words in the text as nodes, rather than a large graph for the en-

tire corpus. In each text, using a very small sliding window, each word in the text is only edgewise connected to its  $p$  adjacent words (including itself, self-joining), rather than fully connected to all word nodes. In this way, the graph size can be reduced from the point of view of nodes and edges to reduce memory consumption. 2) The representation of the same word nodes and the weight of the edges between the same word pairs are shared globally and updated through the message propagation mechanism of text level graph, which eliminates the dependence of single input text and the whole corpus and supports online testing for new text classification.

### 3.2.1. Building a Text Network Diagram

For a text  $T = \{r_1, r_2, \dots, r_n\}$  containing  $n$  words, in this paper,  $r_i$  is a globally shared  $D$ -dimension word embedding matrix generated by the word vector initialized by BERT. The initial representation of each word node is queried from this embedding matrix, which is updated as model parameters in the training process.

Construct a graph for each input text, regard the words in the text as nodes, and the relations between words as edges. Each word node is connected with  $p$  adjacent nodes on its left and right, and self-connects with itself. The graph of input text  $T$  is expressed as:

$$N = \{r_i \mid i \in [1, l]\} \quad (2)$$

$$E = \{e_{ij} \mid i \in [1, l]; j \in [i-p, i+p]\} \quad (3)$$

where,  $N$  represents the word node set,  $E$  represents the edge set, and  $p$  represents the number of nodes adjacent to each word. Each word representation and the edge weight between words come from two different global shared matrices respectively and are updated as model parameters in the training process. In addition, the edges that occur less than  $k$  ( $k = 2$ ) in the training set are uniformly mapped to a common edge so that the parameters can be fully trained. For a single text example, “Men who commit violence against women will never be forgiven,” the network diagram is structured in **Figure 2**.

In this diagram,  $|V|$  said the training focus on the number of words, for this example, we set  $p = 2$  for the node “violence” and  $p = 1$  for the other nodes. The global shared matrix with all parameters is shown on the right.

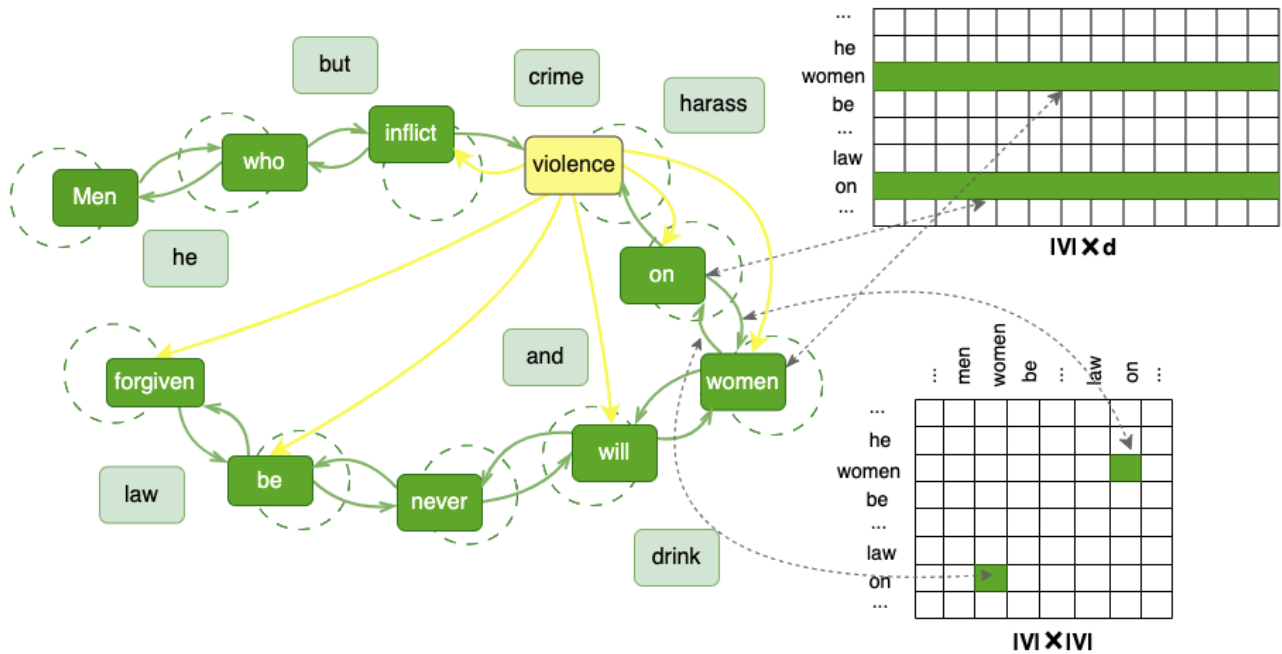
### 3.2.2. Message Passing Mechanism

For each word node in the text graph, MPM collects the information of its adjacent  $p$  nodes, and then updates the node based on the last representation of the node. The process of node updating is expressed as:

$$M_n = \max_{a \in N_p^n} e_{an} r_a \quad (4)$$

$$r'_n = (1 - \eta_n) M_n + \eta_n r_n \quad (5)$$

where,  $M_n \in R^d$  represents the information obtained by node  $n$  from one of its neighboring nodes  $a$ , that is, the representation of each neighboring node  $a$



**Figure 2.** “Men who commit violence against women will never be forgiven” text network diagram structure.

multiplied by the weight of the edges between them  $e_{am} \in R^1$ . Then, the domain node information collected by each node is maximized to obtain a new D-dimensional representation.  $r_n$  represents the last representation of node  $n$ ,  $\eta_n \in R^1$  represents the retained information in  $r_n$ , and  $r'_n$  represents the updated representation of node  $n$ .

Finally, all nodes in a text graph use the softmax function to predict the emotional label of the target text:

$$Z_i = \text{softmax}\left(\text{Relu}\left(W \sum_{n \in N_i} r'_n + b\right)\right) \tag{6}$$

where,  $W \in R^{d \times c}$  is the vector representing the parameter matrix mapped to the output space,  $N_i$  is the node set of text  $i$ , and  $b \in R^c$  is the bias.

The loss function is the cross entropy loss of all texts, defined as follows:

$$\text{loss} = -g_i \log Z_i \tag{7}$$

where,  $g_i$  represents the one-hot vector of the real label, and  $Z_i$  represents the category probability value of model output. The objective of the loss function is to minimize the cross entropy loss of the model prediction and the real label.

## 4. Emotion Analysis of Microblog Comments on “Beating Someone in Tangshan Barbecue Restaurant” Incident

### 4.1. Data Source

In this paper, the distributed Python Weibo\_crawler was used to collect content related to topics in Sina Weibo with the keyword of “Hitting people in Tangshan Barbecue restaurant”. The data attributes crawled were user id, user nickname, microblog text, Etter users, topic, number of reposts, comments, likes and bid.

After data cleaning, invalid comment content (including blank content, repeated content and obviously illogical language text) was eliminated, and 40,788 pieces of data with comment content were obtained. Therefore, firstly, the machine algorithm is compared with the emotion classification marked by human, and a more accurate positive and negative text corpus is obtained by further adjustment. Since the event studied in this paper is social violence, netizens' comments usually have obvious emotional tendency, so only the emotion dichotomy (positive and negative) is studied. Positive comments are represented by 1, while negative comments are represented by 0. Data set 8:1:1 is divided into training set, verification set and test set (**Table 1**).

## 4.2. Experimental Environment and Parameter Setting

The initialization word embedding part of this paper uses the "Bert-Base-Chinese" Chinese pre-training model published by Google. This model is stacked with 12 layers of Encoder layer of Transformer, the hidden layer dimension is 768, the activation function is Relu, and the parameter of multiple Attention mechanism is 12. The total parameter size of the model is 110 MB. The TextLevelGCN section creates a text level graph with a slide window size of 4 and an optimizer of AdamW. The Alpha value was used to balance the feature fusion output of BERT and TextLevelGCN respectively, with the weight set to 0.7. (**Table 2** and **Table 3**).

**Table 1.** Sample microblog comment data.

Microblog text	Label
When can women get real protection and respect from society, and I hope a fair outcome.	1
If today I look on coldly, there will be no one to shout for me when disaster comes to me.	1
What the hell is going on with Operation Tangshan Thunder? This half a month of posturing is actually in order to justify the mouth of the net friend?	0
I'm relieved to see the whole web cursing those scum at the grill. I wish you are in jail!	0

**Table 2.** Experimental environment configuration.

Experimental Environment	Labelconfiguration Data
The programming language	Python3.7
.Development tool	Pycharm
Operating system	MacOS Mojave10.14.6
Deep learning framework	Pytorch
Processor	1.8 GHz Intel Core i5



**Table 3.** Parameter setting.

Parameter	Parameter value
Alpha	0.7
Epochs	60
Batch_Size	128
Dropout	0.5
Learning_Rate	1e-3
Hidden_Size	768
Max_Length	128
Optimizer	AdamW
LossFunction	Cross Entropy
N_Gram	4

### 4.3. Model Evaluation Index

In order to verify the effect of the model in the task of text emotion classification, this paper takes Accuracy, Precision\_score, Recall\_score and MF1-score as four evaluation indicators.

Mf1-score, combined with Precision and Recall, can reflect classification performance more comprehensively. Only when P and R are equal to 1, MF1 reaches its maximum value. When evaluating the performance of the classifier, the closer the value of MF1 is to 1, the better the performance of the classifier is.

### 4.4. Model Results and Comparative Analysis

In order to verify the improvement superiority of the BERT-TextLevelGCN model, the model in this paper is compared with the following baseline model: (Table 4).

Comparing and analyzing the performance of different models, the BERT-TextLevelGCN feature fusion model used in this paper has the best overall performance, with an accuracy of 0.8723 and an accuracy of 0.9011. The recall rate reached 0.7977. As can be seen from Table 4, TextLevelGCN has higher model classification accuracy than TextGCN, which indicates that building a separate Text graph for each text and adding the function of message propagation mechanism can obtain more robust text features. The classification accuracy of BERT-GCN is higher than that of Text-GCN, which indicates that the addition of pre-training model can greatly improve the semantic representation ability of the initial Text vector, thus enhancing the effect of semantic feature fusion. By comparing the results of the feature fusion model of Bert-TextLevelGCN in this paper with BERT and TextLevelGCN, we can see that compared with BERT, the accuracy is increased by 0.0481, F1 is increased by 0.055, and the accuracy is increased by 0.0228 compared with TextLevelGCN. F1 improved by 0.1302. This shows that the interpolation update feature fusion strategy in this paper can better integrate the information of text graph with the semantic information after BERT pre-training.

**Table 4.** Model performance comparison.

Model	Acc	$P_i$	$R_i$	MF1
BERT	0.8242	0.8461	0.7921	0.8041
TextGCN	0.7588	0.7675	0.7656	0.7666
BERTGCN	0.7820	0.7897	0.7762	0.7829
TextLevelGCN	0.8495	0.7709	0.7666	0.7696
<b>BERT-TextLevelGCN</b>	<b>0.8723</b>	<b>0.9011</b>	<b>0.7977</b>	<b>0.8349</b>

## 5. Concluding Remarks

In this paper, a fusion model based on pre-training model BERT and graph neural network method TextGCN method is proposed for Sudden social security violence with obvious emotion. On the one hand, TextGCN is improved by first constructing text-level Text subgraph, building the text subgraph at the text level separately, introducing message propagation module. The feature information of domain node aggregation is adjusted by word weight maximum pooling. On the other hand, the feature information containing rich contextual semantics obtained by BERT is weighted with the semantic features obtained by TextLevelGCN.

As a weight combination model, the computational complexity of the Bert-TextLevelGCN model in this paper has increased, but the experimental results verify the accuracy and effectiveness of the prediction of this model, indicating that the model combines BERT well and can capture sufficient contextual semantic information. TextLevelGCN has the advantages of flexible modeling and information fusion, which enhances the network's ability to extract text features, greatly improves the classification performance of the model, and saves running memory by reducing the sliding window. However, this paper still has some shortcomings, including too long training time for the model. The reason is that BERT model has a large number of parameters and many model layers. In the process of fine-tuning downstream tasks, a large number of parameters need to be updated by back propagation, which consumes more time.

## Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

## References

- [1] (2008) Emergency Response Law of the People's Republic of China. Bulletin of the Supreme People's Procuratorate of the People's Republic of China. No. 1, 5-13.
- [2] Chen, J.H. and Li, G. (2016) A Survival Analysis of the Evolution of Online Public Opinion of Emergency Social Security Events: Based on the Analysis of 70 Major Social Security Events. *Journal of Information*, **35**, 70-74.
- [3] Aggarwal, C.C. and Zhai, C.X. (2012) A Survey of Opinion Mining and Sentiment

Analysis, *Mining Text Data*, Springer, Boston, 415-463.

[https://doi.org/10.1007/978-1-4614-3223-4\\_13](https://doi.org/10.1007/978-1-4614-3223-4_13)

- [4] Yu, T.R., Jin, R., Han, X.Z., Li, J.H. and Yu, T. (2020) A Review of Pretraining Models for Natural Language Processing. *Computer Engineering and Applications*, **56**, 12-22.
- [5] Chen, Y.H. (2015) Convolutional Neural Network for Sentence Classification. UWSpace. <http://hdl.handle.net/10012/9592>
- [6] Peng, H., Li, J.X., *et al.* (2018) Large-Scale Hierarchical Text Classification with Recursively Regularized Deep Graph-CNN. In *Proceedings of the 2018 World Wide Web Conference (WWW'18)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 1063-1072. <https://doi.org/10.1145/3178876.3186005>
- [7] Kipf, T.N. and Welling, M. (2016) Semi-Supervised Classification with Graph Convolutional Networks. arXiv: 1609.02907.
- [8] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P. and Bengio, Y. (2017) Graph Attention Networks. arXiv: 1710.10903.
- [9] Salha, G., Hennequin, R. and Vazirgiannis, M. (2019) Keep It Simple: Graph Autoencoders without Graph Convolutional Networks. arXiv: 1910.00942.
- [10] Kolda, T.G., Pinar, A., Plantenga, T. and Seshadhri, C. (2014) A Scalable Generative Graph Model with Community Structure. *SIAM Journal on Scientific Computing*, **36**, C424-C452. <https://doi.org/10.1137/130914218>
- [11] Yan, S., Xiong, Y. and Lin, D. (2018) Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, **32**. <https://doi.org/10.1609/aaai.v32i1.12328>
- [12] Yao, L., Mao, C. and Luo, Y. (2019) Graph Convolutional Networks for Text Classification. *Proceeding of the AAAI Conference on Artificial Intelligence*, **33**, 7370-7377. <https://doi.org/10.1609/aaai.v33i01.33017370>
- [13] Peters, M., Neumann, M., Iyyer, M., *et al.* (2018) Deep Contextualized Word Representations. *Proceedings of NAACL*, **9**, 2227-2237.
- [14] Wu, X.D. (2021) Research on Text Classification Model Based on Improved Graph Neural Network. Northwestern University, Evanston.
- [15] Devlin, J., Chang, M.W., Lee, K., *et al.* (2018) Bert: Pre-Training of Deep Bidirectional Transformers for Language Understanding. arXiv: 1810.04805.