


# TGCN-Bert Emoji Prediction in Information Systems Using TCN and GCN Fusing Features Based on BERT

Zhangping Yang, Hongqing High-tech Industrial Park, China\*

 <https://orcid.org/0009-0009-2810-899X>

Xia Ye, Hongqing High-tech Industrial Park, China

Hantao Xu, Hongqing High-tech Industrial Park, China

## ABSTRACT

In recent studies, graph convolutional neural networks (GCNs) have been used to solve different natural language processing (NLP) tasks. However, few researches apply graph convolutional networks to short text classification. Emoji prediction, as a complex sentiment analysis task, has received even less attention. In this work, the authors propose TGCN-Bert which combines pre-trained BERT temporal convolutional networks (TCNs) and graph convolutional networks for short text classification and emoji prediction. They initialize the nodes with the help of BERT and define the edges in text graph based on the term frequency-inverse document frequency (TF-IDF) and positive point-wise mutual information (PPMI). They employ the model for emoji prediction task, and a metric based on emoji clustering is developed to better measure the validity of emoji prediction results. To validate the performance of TGCN-Bert, they compare it with other GCN variants on short text classification datasets and emoji prediction datasets; experiments show that TGCN-Bert achieves better performance.

## KEYWORDS:

Emoji Prediction, Graph Convolutional Network, Sentiment Analysis, Text Classification, BERT, Emoji Clustering

## 1. INTRODUCTION

With the rapid development of machine learning, more and more projects are gradually focusing on everyday life. Machine learning has achieved a lot in fraud detection (Almomani, A., et al., 2022; Gaurav, A., et al., 2022), traffic management (Liu, R., et al., 2022; Jain, A., et al., 2022), and even big data (Stergiou, C., et al., 2021; Lv, L., et al., 2022; Xu, Z., et al., 2023; Barbosa, A., et al., 2022). In the field of natural language processing, Yen et al. (2021) identify social user status by analyzing post content and social behavior. Gu et al. (2022) utilize a contextual Word2Vec model to understand

DOI: 10.4018/IJSWIS.331082

\*Corresponding Author

This article published as an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

the words out of vocabulary in Weibo corpus. Sentiment analysis of social users' speech can reveal users' views on current hot events and control public emotions. Alowibdi et al. (2021) discovered that COVID-19 has propagated fear, anxiety, hope, and other emotions throughout the general population by examining the content of Twitter users' messages. In order to identify conflicts in trending Twitter topics, Al-Ayyoub et al. (2018) suggest a hybrid strategy. This is done in order to examine the dynamics of online groups and the behavior of their interactions. The crucial factors that influence tourists' pleasure were identified by Bai et al. (2023) by examining the comments posted by tourists. This discovery is extremely important for the study of consumer behavior. More and more social media data is distributed in unstructured forms (Singh, S. K., & Sachan, M. K., 2021), such as emoji, which can help sentiment analysis. As a kind of emotional symbols, emojis are frequently used with natural language, especially in social media posts such as Weibo(in China) and Twitter. There are thousands of icons that can express emotions. And there are many variants of emojis that have similar meaning but have different features such as skin tones and image amounts, etc. Still, this does not affect how well emojis combined with short text can express the emotions of users. Evans (2017) sees nonverbal cues as an expression of emotions. However, in traditional digital communication, these cues may get lost, which can lead to communication bias. Nevertheless, emojis fulfill this function. To some extent, emojis even reflect the real emotional polarity of the authors. It may lead to the opposite emotional polarity via text without emojis. According to Na'aman et al. (2017), emojis can serve as syntactic components in the text in the same way words do. But on the other side, emojis may also mislead humans. Each individual has a different perception of emotions containing emojis, which is known as cognitive bias (Miller, H., et al., 2017). Luckily, short text combined with emojis can express emotions that fit what most people perceive. The applications of emoji prediction are far-reaching. Social media platforms can analyze user opinions on hot topics through the sentiment of social posts, while commercial platforms can improve product quality and service experience based on user feedback. In addition, for intelligent question-answering(QA) models like ChatGPT, the abstract understanding and actual use of emoji can also improve the performance of the model on more tasks. Thus, prediction of emojis would be beneficial for achieving better language understanding (Barbieri, F., Ballesteros, M., & Saggion, H., 2017).

The emoji prediction task was originally proposed in (Kralj Novak, P., 2015). The task aims to find the most appropriate emoji according to a short piece of text. As everyone knows, it has become a trend that more and more natural language processing(NLP) task datasets contain emojis. In general, it would be helpful to pre-train models by predicting emojis to boost their performance on other NLP tasks. And the knowledge learned via emoji prediction tasks can be well transferred to solve other NLP tasks e.g., emotion prediction, sentiment analysis and sarcasm detection (Felbo, B., 2017). The fine-tuned models pretrained on emoji prediction datasets tend to perform better.

Essentially, emoji prediction is a more complex text classification task. In NLP field, many classic deep learning models, such as CNN (Kim, Y., 2014) and RNN (Hochreiter, S., & Schmidhuber, J., 1997), have been applied to short text classification tasks. However, the former is unable to process massive amounts of data in parallel and has pretty high computational complexity. Bai et al. (2018) argued that convolutional networks should be considered one of the main candidates when modeling sequence data, and they proposed temporal convolutional networks(TCNs), a model that has the advantages of CNN and RNN and is expected to replace RNN in some aspects. Graph neural networks(GNNs) (Cai, H., et al., 2018; Battaglia, P., 2018) have a powerful ability to process graph structure data and reach great achievement in many NLP fields (Liu, X., et al., 2018; Zayats, V., & Ostendorf, M., 2018). And GNN, especially graph convolutional network(GCN) applied to text classification (Kipf, T. N., & Welling, M., 2016) has attracted great attention. Yao et al. (2019) proposed TextGCN which has the capacity to build a single text graph of a corpus with the help of document word relations and word co-occurrence. Huang et al. (2019) built text-level graphs for each text instead of a single graph for the whole corpus, which significantly reduces the time overhead of

training GCNs. However, some GCN models perform worse than classic models on the short text datasets and sentiment analysis datasets.

In this paper, we propose a novel deep learning method for solving short text classification and take a fresh trial to apply GCN and TCN to the emoji prediction task. We design a neural network for parallelly encoding short text. From the perspective of GCN, the text graph can capture both document-to-word and global word-to-word relationships. From the view of the other component, TCN has flexible receptive field size and stable gradients and allows variable-length inputs. Specially, following BertGCN and RobertaGCN (Lin, Y., et al., 2020), inspired by the ideas of combining GCNs and pre-trained language models (PLMs) (Devlin, J., et al., 2018; Liu, Y., et al., 2019; Yang, Z., et al., 2019; Brown, T., et al., 2020), we build a global text graph according to corpus. Considering semantic information in node representation and sequence order, which TextGCN ignores. Then we utilize the pretrained vectors from BERT to obtain high-quality contextual information. A layer of cross-attention mechanism is applied to the feature vectors from GCN and TCN to achieve feature fusion.

The contribution of this work can be summarized as follows:

- (1) A novel neural network consisting of GCN and TCN is introduced for solving short text classification. To validate the effectiveness of the proposed method, several short text classification datasets are used to evaluate it. The experiments show that the proposed model achieves the state-of-the-art effect on almost all datasets.
- (2) Unlike most methods based on traditional deep learning neural networks, the proposed GCN-based model are first tried to be applied to the emoji prediction task.
- (3) Inspired by Ma et al. (2020), an emoji clustering method is designed. And evaluation rules are redefined, which makes emoji prediction results more reasonable.

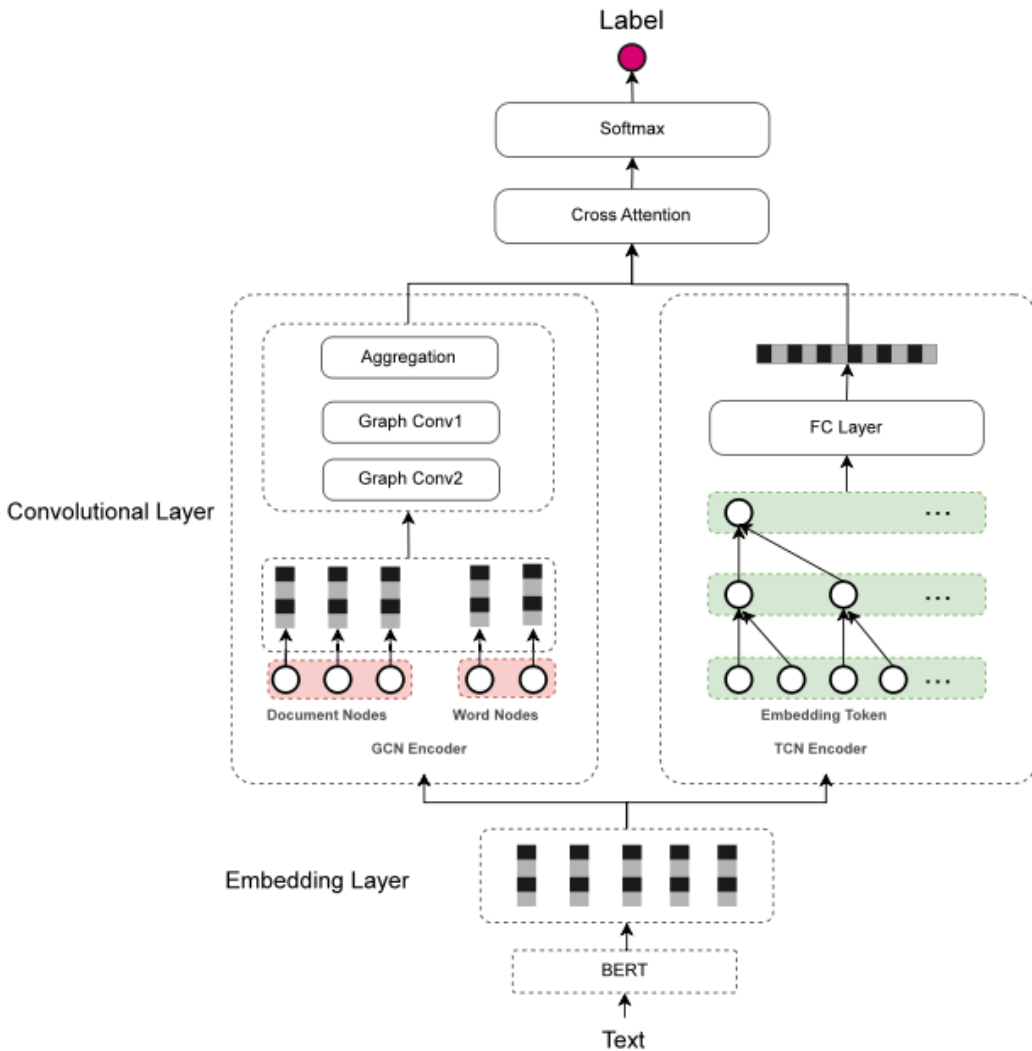
## 2. RELATED WORK

In recent years, some studies have been proposed to solve the emoji prediction task. Initially, most models based on traditional machine learning methods were introduced to solve the emoji prediction task in SemEval Task 2 (Barbieri, F., et al., 2018). Specifically, Baziotis et al. (2018) used a Bi-LSTM with attention mechanisms and pretrained word2vec vectors. Considering the impact of additional information on the sentiment of the tweet, they also used external resources for associating each input text with information about emotions, familiarity, and concreteness. Felbo et al. (2017) built Deepemoji based on two stacked Bi-LSTM with skip connections and attention mechanisms for better understanding text at character-level granularity. Zhang et al. (2020) proposed utp-BiLSTM and made full use of other user tags such as time and location in tweets, supplemented by a multi-head attention mechanism to predict emojis. Based on Deepemoji, Barbieri et al. (2018) designed a label-wise attention Bi-LSTM whose attention mechanism modules focused on specific labels.

However, with the advent of pre-trained language models, fine-tuning has become a paradigm for solving NLP tasks. Ma et al. (2020) collected and produced broader emoji datasets with a greater number of labels. They fine-tuned BERT on self-collected emoji datasets, and the models performed well on all datasets. The excellent performance may be attributed to the text comprehension capacity of BERT. In the research (Gupta, A., et al., 2021), researchers employed BERT on emoji recommendations by using time and location parameters.

It is noted that GCN is proposed in (Kipf, T. N., & Welling, M., 2016) and achieves state-of-the-art classification results on a great number of graph datasets. It has become a trend to process natural language corpora into graph structure data and depend on GNNs to cope with NLP tasks such as text classification and sentiment analysis. TextGCN (Yao, L., Mao, C., & Luo, Y., 2019) builds a global text graph for the whole corpus and updates the nodes by using the full batches. In general, documents and sentences are regarded differently. Some researchers treat documents and sentences

Figure 1. The overview of TGCN-Bert



as graphs consisting of word nodes, while others treat documents and sentences as nodes. And as the amount of data increases, the scale of the global text graph will also increase dramatically, which may lead to extremely high memory consumption. In order to alleviate the memory consumption when constructing the global graph, GNNs are used for inductive text classification experiments. Huang et al. (2019) designed a text-level GNN architecture that builds a text graph of each input sentence. In each input text, words in the text are connected by several words nearby in a specific sliding window. And the representation of word nodes and the edges between word pairs share the same weight globally. Considering the advantages of PLMs, works that combine GNNs and PLMs emerge gradually (Lu, Z., Du, P., & Nie, J., 2020; He, Q., Wang, H., & Zhang, Y., 2020; Lin, Y., et al., 2021). BertGCN (Lin, Y., et al., 2021) complements BERT's textual semantic features by initializing and updating nodes with the help of pre-trained word embeddings.

Our work is inspired by the work of adapting GNNs and BERT into text classification (Ye, Z., et al., 2020; Lin, Y., et al., 2021) and the demands of new methods of evaluation in the emoji

prediction task (Ma W., et al., 2020). Different from these works, we adapt GNNs and PLMs to the emoji prediction task. And we propose a new evaluation metric for emoji prediction. Besides, a dual-channel neural network structure combining TCN and GCN is designed. Two neural networks have their own advantages and could complement each other from different perspectives.

### 3. METHOD

We follow the design ideas of BertGCN. The overview of TGCN-Bert is shown in Fig. 1 In the channel consisting of GCN, we initialize representations for document nodes using BERT. During the training process for GCN, these representations get updated. And we will obtain the dense vectors from the last hidden layer of GCN. In the other channel, the [CLS] token embeddings will be fed into the TCN module. And we can obtain the output vectors from TCN as well. Finally, the interdependence between the two feature vectors will be extracted by calculating the cross-attention score. And the output will be sent to a softmax layer for predictions. In this measure, the model is able to understand social texts from different perspectives.

#### 3.1 Notations

The notations that will be used in the paper are listed as follows:

#### 3.2 BERT Representations

In order to improve the model performance, we initialize the nodes by using BERT. We refer to the results of BERT in text classification and use them as part of the joint probability. BERT's high-quality pre-trained word vectors can represent contextual semantics, and using these word vectors to initialize word nodes can enable the model to achieve better classification performance.

#### 3.3 Text Graph Construction and Graph Convolution

Like TextGCN (Yao, L., Mao, C., & Luo, Y., 2019) and BertGCN (Lin, Y., et al., 2021), we take the same approach to building heterogeneous graphs containing document nodes and word nodes. Specifically, we employ the term frequency-inverse document frequency(TF-IDF) and positive point-wise mutual information(PPMI) to define edges between nodes. The former counts the times of appearance of words by term frequency and demonstrates the weight of words by inverse document frequency which is the logarithmically scaled inverse fraction of the number of documents that contain the words. The latter can measure the associations between words. The weight of the edge between node  $i$  and node  $j$  is defined as below:

$$A_{i,j} = \begin{cases} \text{TF-IDF}(i, j), & i \text{ is document, } j \text{ is word} \\ \text{PPMI}(i, j), & i, j \text{ are words and } i \neq j \\ 1, & i = j \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

For the node feature matrix, we use the embeddings of [CLS] token to represent the document node embeddings  $X_{doc}$ . And the node feature matrix is initialized in (2), where  $n$  is the number of nodes and  $d$  is the dimensionality of the embeddings.

$$X = \begin{pmatrix} X_{doc} \\ 0 \end{pmatrix}_{n \times d} \quad (2)$$

## Notations

TF-IDF	A statistical method that calculates the importance of a word in a document by its frequency in different documents
PPMI	A statistical method used to measure the correlation between words
$A_{i,j}$	Matrix of weights between nodes
$X_{doc}$	Matrix of document embedding vectors for all document nodes
$X$	Matrix of node features
$L^{(i)}$	Mathematical representation of the $i$ -th graph convolutional layer
$\tilde{A}$	Normalized symmetric adjacency matrix of $A$
$W$	A weight matrix of the layer
$\rho$	An activation function of the layer
$D$	Matrix of the degree of all nodes
$\tilde{D}$	Normalized symmetric adjacency matrix of $D$
$O_{GCN-Bert}$	Feature vector output of the GCN-Bert channel
$g$	An activation function of the output layer of GCN-Bert channel
$x_0, \dots, x_T$	Text sequence as input to model
$y_0, \dots, y_T$	Sequence of predictions output by the model
$x_0, \dots, x_t$	Sequence of text inputs before time step $t$
$x_{t+1}, \dots, x_T$	Sequence of text inputs after time step $t$
$F(s)$	Output of the dilated convolution on input text sequence
$d$	Factor of dilated convolution
$k$	Size of convolutional filter
$O_{TCN-Bert}$	Feature vector output of the TCN-Bert channel
Cross( $;$ , $\cdot$ )	Cross attention mechanism
$Z$	Model prediction probability

The matrix  $X$  will be treated as input to the GCN module. And the node feature matrix of the  $i$ -th GCN layer  $L^{(i)}$  is computed as:

$$L^{(i)} = \rho(\tilde{A}L^{(i-1)}W^{(i-1)}) \quad (3)$$

Where  $\tilde{A}$  calculated in (4) is a normalized symmetric adjacency matrix of  $A$ , which allows fast execution of sparse matrix operations and avoids gradient vanishing or gradient exploding.  $\rho$  is an activation function. And  $D$  represents the degree of the node  $i$ , which is calculated in (5). And the outputs of the GCN-Bert channel can be defined in (6).

$$\tilde{A} = \tilde{D}^{-\frac{1}{2}}(I + A)\tilde{D}^{-\frac{1}{2}} \quad (4)$$

$$D_{ii} = \sum_j A_{ij} \quad (5)$$

$$O_{GCN-Bert} = g(X, A) \quad (6)$$

### 3.4 Temporal Convolution

Like other sequence modeling tasks, TCN is given an input sequence  $x_0, \dots, x_T$ , and wishes to predict the corresponding output sequence  $y_0, \dots, y_T$  at each time. However, different from other networks, the key constraint is that TCN is constrained to only use those observed inputs while predicting the output  $y_t$  at time  $t$ . For the output  $y_t$ , the causal constraint is that only  $x_0, \dots, x_t$  but not any future inputs  $x_{t+1}, \dots, x_T$  are used for prediction. And TCN is based upon two rules, one is that TCN produces an output of the same length as the input, and the other is that sequences from the future cannot be used. Based on what is mentioned above, TCN uses causal convolutions, where an output at time  $t$  is convolved with elements from time  $t$  and earlier in the previous layer.

In order to enable causal convolution to have a longer memory, dilated convolution is introduced to enable an exponentially large receptive field. The dilated convolution operation on elements of the sequence is defined in (7).

$$F(s) = \sum_{i=0}^{k-1} f(i) \cdot x_{s-d \cdot i} \quad (7)$$

Where  $F$  stands for the operation,  $s$  is the elements of the sequence  $d$  is the dilation factor,  $k$  is the size of the filter, and  $s - d \cdot i$  accounts for the direction of the past. Additionally, a residual block (He, K., et al., 2016) is employed in TCN. The final output vector after using any activation function  $g$  is shown in (8).

$$O_{TCN-Bert} = g(x + F(x)) \quad (8)$$

### 3.5 Interpolating GCN, TCN, and BERT Predictions

After the parallel operation of both neural network channels, we obtain respective outputs from two channels. For the purpose of taking the respective advantages of GCN-Bert and TCN-Bert channels, feature vectors of output are fused in a fusing layer within cross-attention mechanism. Finally, the prediction labels can be obtained through final one dense layer with softmax activation. In addition, we find that adding an auxiliary classifier by feeding document embeddings to a dense layer also helps optimize the whole model and improve performance, which is expressed as (9).

$$Z_{BERT} = \text{soft max}(WX) \quad (9)$$

Specially, the final prediction is the interpolation of the prediction from TGCN-Bert and BERT, which is given by:

$$Z = \text{soft max}(W \cdot \text{Conv}(\text{Cross}(o_{GCN-Bert}, o_{TCN-Bert}) + b) + Z_{BERT} \quad (10)$$

## 4. EXPERIMENTS

### 4.1 Experimental Setup

We conduct validation experiments on five widely-used text classification benchmarks, respectively: 20NG, R8, R52, Movie Review(MR) and Ohsumed. We aim to validate the graph structure data processing capacity of TGCN-Bert and compare it with other GCN variants.

20NG collects newsgroup documents evenly divided into 20 different topics. R8 and R52 are both subsets of the Reuters 21578 datasets. MR dataset consists of short movie reviews for binary sentiment classification. And Ohsumed is extracted from MEDLINE database, which is designed for multi-label classification, however, texts with only one label are retained.

To apply our model to emoji predictions, emoji datasets from research (Barbieri, F., et al., 2017) are employed. In total, there are about 500,000 tweets in the dataset, which are divided into 20 labels, and each tweet is labeled with a single emoji tag in a distant supervision manner (Mintz, M., et al., 2009).

### 4.2 Validation Experiments

Since the proposed neural network structure contains GCN, TGCN-Bert also has the capacity to process graph structure data. In order to verify the effectiveness of TGCN-Bert in text classification tasks, we run experiments on five widely used text classification benchmarks. Several GCN models and pretrained models, which show excellent performance, are chosen to be baselines. Specifically, we compare our proposed model to TextGCN (Yao, L., Mao, C., & Luo, Y., 2019), text-level GCN (Huang, L., et al., 2019), BertGCN (Lin, Y., et al., 2021) and BERT. TextGCN builds the weights of the global text graph with TF-IDF and PMI for the whole corpus and updates the whole text graph in full batches. Text-level GCN builds a text graph for each input text and establishes a connection with nearby word nodes, which means that constructing text-level graphs can consume less GPU memory. The models above only take advantage of structural features but lose the semantic features of the text. BertGCN initializes the nodes by using the pre-trained word vectors from BERT, which helps the GCN module learn semantic feature expression when it aggregates features. And BertGAT, a variant of BertGCN, employs the graph attention mechanism in addition. And RoBERTaGCN and RoBERTaGAT are trained and initialized with RoBERTa.

To keep the same experiment setting as TextGCN and BertGCN, we employ the same method to process data. For the BERT classifier, we treat the output embeddings of [CLS] token from BERT<sub>base</sub> as the document embeddings that will be sent to a fully connected layer to obtain prediction labels.

Table 1 shows the test accuracy of GCN models and pretrained models. Even if GCN models are utilized with the ability to process graph data, some GCN variants cannot outperform BERT and RoBERTa thoroughly. For one reason, the model like text-level GCN builds a local graph for each input text instead of a global graph, which aims at operating inductive text classification and reducing memory consumption. And it can be predicted that the cost of reducing the size of the text graph is undoubtedly the loss of model accuracy. For another reason, PLMs store more prior knowledge about



Table 1. Results for GCN models and PLM models on text classification datasets

Model	20NG	R8	R52	MR	Ohsumed
TextGCN	86.3	97.1	93.6	76.7	68.4
BERT	85.3	97.8	96.4	85.7	70.5
RoBERTa	83.8	97.8	96.2	89.4	70.7
Text-level GCN	62.3	97.7	94.4	69.9	69.4
BertGCN	89.2	97.9	<b>96.7</b>	86.1	72.8
RoBERTaGCN	89.5	98.2	96.1	89.7	72.8
BertGAT	87.4	97.8	96.3	86.6	71.5
RoBERTaGAT	87.4	97.8	96.5	86.5	71.2
TGCN-Bert	90.5	<b>98.6</b>	96.7	88.9	73.2
TGCN-RoBERTa	<b>90.8</b>	98.5	96.6	<b>90.3</b>	<b>73.5</b>

natural language understanding(NLU) during the process of pretraining. And this will help improve the performance of the model.

Moreover, pre-trained language models like BERT and RoBERTa are better at dealing with short texts. There is little difference in the performance of all models on the R8 and R52 datasets, which indicates that for the short text news classification task, the current model method has been able to solve the task well after sufficient training. On 20NG and Ohsumed datasets, BertGCN and RoBERTaGCN are significantly better than BERT and RoBERTa, which is because the average text length of the two datasets is much longer than that of other datasets. This also proves that the high-quality text representation of the pre-trained language model can effectively improve the performance of the model after the introduction of the GCN model. From this perspective, combining GCN and pre-trained language models is a novel and practical idea for solving text classification tasks.

By comparing the performance of TGCN-Bert with BertGCN and RoBERTaGCN on each dataset, it can be found that the proposed model achieves the most advanced results on 20NG, R8, MR and Ohsumed datasets, and there is little difference in accuracy compared with other models on R8 and R52. This is because the accuracy of the comparison model on R8 and R52 data sets has exceeded 95%. News classification is based on objective facts, and the text features of this kind of data set are obvious. As a binary classification sentiment analysis dataset, MR has simple labels, and its content is short movie reviews of certain movies. GCN can be used to mine the association between movies and sentiment words so as to improve the effect of the model. On 20NG and Ohsumed datasets, BertGCN, RoBERTaGCN and the proposed model also perform quite well. The reason may be that there are many professional terms involved in the dataset, and GCN represents them as graph structure data, which is associated with the classification label, making it easier for the model to learn the relationship between some important words and labels. The TGCN-Bert and TGCN-RoBERTa models are better than other comparison models. This is because, compared with BertGCN and RoBERTaGCN, the TCN module allows the model to encode text vectors with flexible receptive fields, and the model effect will be further improved. For the choice of pre-trained language model, the use of BERT or RoBERTa is relevant for downstream tasks.

### 4.3 Emoji Clustering

Emoji prediction task aims to select the most appropriate emoji for a tweet. Essentially, the task is a multi-class text classification problem. Therefore, using metrics such as accuracy and F1 score to evaluate models seemed rational in previous research. However, due to the significant differences in emoji usage habits across different Twitter user groups, using traditional metrics is

not satisfactory and may even result in low accuracy. Emojis have spawned thousands of variants and we can find that some emojis with similar synonyms are often used almost interchangeably. Barbieri et al. (2018) take this into account and additionally use accuracy@5 and coverage error to evaluate emoji prediction models. The former calculates the accuracy of the five prediction labels with the highest probability including the true label. In other words, in a single text emoji prediction process, if the first five predictions predicted by the model contain the real label, then the prediction is fairly correct. The latter metric describes the relative distance at which the predicted label recovers from the real label. However, accuracy@5 is just a rough extension of the reasonable metrics of emoji predictions. In the top 5 predictions, if there are emojis with large sentiment differences, it indicates that the model has not really understood the semantics of emoji. However, the evaluation metric proposed in this paper is based on the emoji clustering results, which are grouped according to the correlation between emoji. Using this evaluation metric, we cannot only measure whether the model really understands the text sentiment, but also enhance the interpretability of the evaluation metric.

Considering the problems we encounter, we propose a new accuracy metric based on emoji clustering (Acc@C, which is an evaluation metric that calculates the accuracy of predicting the inclusion of a label in a cluster group). Since different emojis can be substituted for each other in many scenes, we cluster emojis from the perspective of image and statistics. We perform K-means clustering on emoji images with setting the initial number of clustering categories to 5. And we repeat the clustering operation several times to reduce the impact of randomness. Then, in order to obtain the appreciate clustering results, we increment the number of clusters and iterate the above clustering process. Nevertheless, the clustering effect based only on emoji images would not be satisfactory. Therefore, we fine-tune BERT with the same experiment setting and evaluation as (Barbieri, F., et al., 2018), which aims to obtain the correlation between different emojis with the help of the confusion matrix. Based on the correlation coefficient between emojis, weakly correlated or negatively correlated emojis are ensured not to be in the same cluster. For example, after emoji clustering based on images, camera images and the US flag are divided into one group, which does not make sense. But according to the correlation coefficient between them, camera emojis and flag emojis are divided into different groups. The normalized confusion matrix is produced by fine-tuned BERT, as is shown in Figure 2.

From Figure 2, it can be seen that the fine-tuned BERT classification results are generally reliable. In the heatmap, shades of color can indicate the predicted distribution. It is worth noting that the predicted classification of the two camera emojis is often confused with each other, which indicates that the two are semantically similar and can be substituted for each other in the tweet text. The same goes for lightly smiling and beaming face. This interesting phenomenon may reveal that these emojis have similar meanings and are strongly correlated in the usage habits of the Twitter user group. This phenomenon also occurs between red hearts and air kisses, which together show affection for someone or something.

In general, the proposed emoji clustering method is reasonable. Emojis contain love, emoticons and other specific images. We can find that the results of specific images are reasonable, and emojis in group *lightly smile* would express similar emotional content and can be used interchangeably in some context. Although a few emojis in the same group may have slightly larger emotional differences. For the purpose of avoiding the phenomenon that there are only one or two emoji(s) per cluster, we allow emojis to appear repeatedly in the clustering results. While the repetition is also a fact that we follow the results of image clustering. The emoji clustering results are shown in Figure 3.

#### 4.4 Emoji Prediction Experiments

In our study, we choose the dataset published in SemEval Task 2 in English for the evaluation of our proposed model. There is a lot of noise in the training data, including #hashtags, @users, retweets, location and colloquial expressions. The noise may adversely affect model performance. To avoid

Figure 2. The heatmap of the normalized confusion matrix

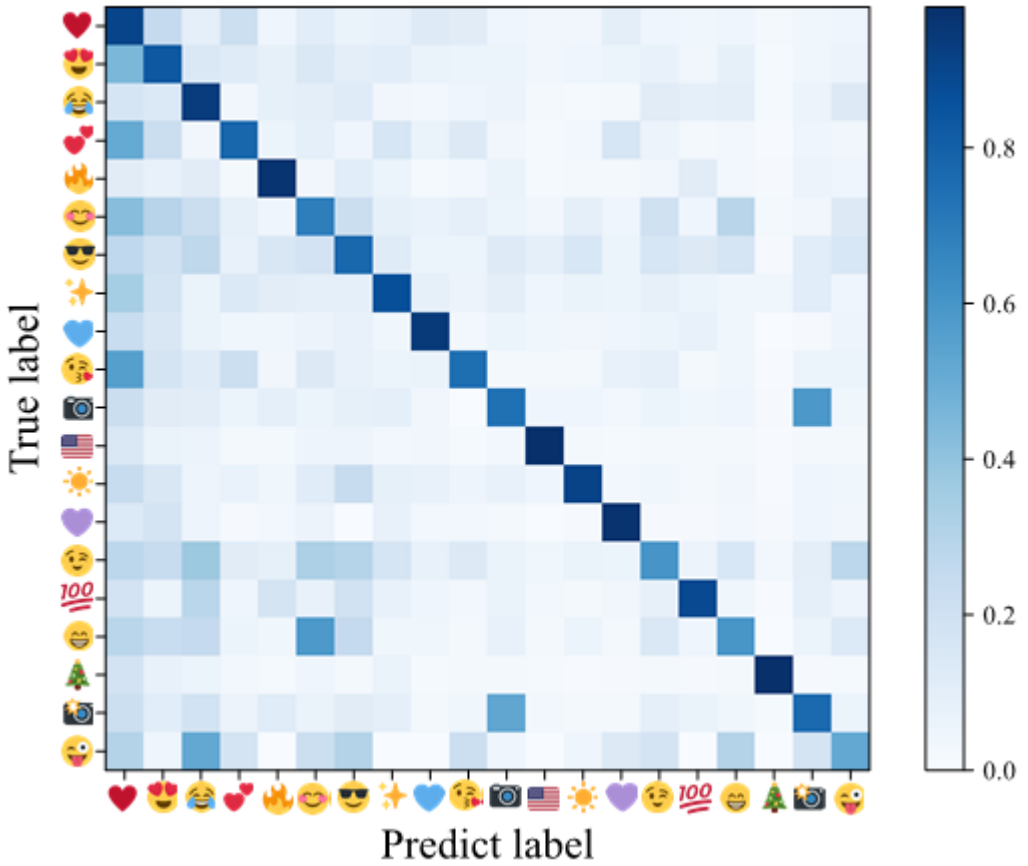
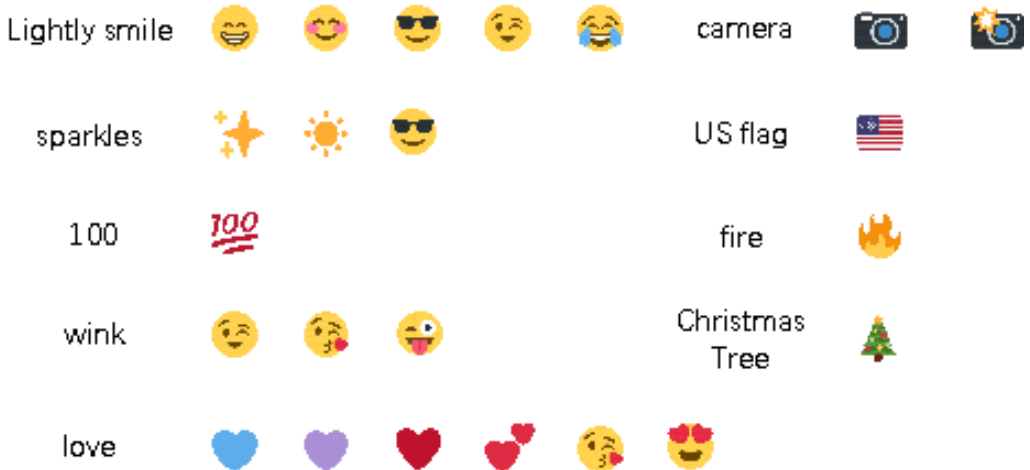


Figure 3. Results of 20-emoji clustering based on images and statistics



it, all the data was preprocessed. We removed @users, retweet tags(RT) and #hashtags. And the irregularly spelled location words were corrected. In order to evaluate the performance of emoji prediction systems, we use accuracy, accuracy@5 and accuracy@C as evaluation metrics. Table 2 shows the results of our model and the baselines in the emoji prediction task. Deepmoji uses a stacked BiLSTM network with an attention mechanism added. 2-BiLSTM<sub>α</sub> uses an attention mechanism based on emoji tags. utp-BiLSTM uses a multi-head attention mechanism, focusing on three parameters: user, time and location.

It can be found in the Table 2 that models using BERT generally perform better because the pretrained language model has learned general knowledge and has an advantage in sentiment analysis. TGCN-Bert performs best in terms of accuracy. Due to the operation of fusing features through TCN and GCN, the model can better extract the sentiment information in the text. And it is worth noting that BertGCN performs worse than BERT model, which may be caused by the fact that GCN is not good at aggregating text features containing complex emotions. Comparing BertGCN and TGCN-Bert, our proposed model has better performance on clustering-based evaluation metrics, which means the proposed model benefits more from fusing features via TCN and GCN channels. But their performances are not as good as BERT's due to the gap between transformer models and GCN models in sentiment text understanding.

## 5. CONCLUSION AND FUTURE WORK

In this work, we novelly propose a two-channel neural network model that takes advantages from BERT, TCNs and GCNs. We efficiently train the GCN channel of TGCN-Bert by using locally stored node information from BERT embeddings and update them. We use TCN to replace BiLSTM or CNN as the encoding layer, and obtain semantic features by means of dilated convolution. In this way, it not only uses the semantic prior knowledge of BERT, but also fuses semantic features from GCN and TCN with different receptive fields. Compared with several mainstream GCN variants, our model achieves fair performance in the short text classification task. In addition, we design a new metric based on emoji clustering. This emoji clustering method not only utilizes the emoji graphics but also utilizes the statistical information based on the confusion matrix generated by BERT. Using this metric, we apply the model to emoji prediction and also achieve satisfactory results. However, TGCN-Bert uses the whole corpus to construct a global text graph and undergoes dual-channel convolutional coding, which incurs a large time overhead for initialization and training. And in this work, we use document statistics to build a global text graph, which might lead to sub-optimal performance. In addition, the newly proposed evaluation metric Acc@C refers to the confusion matrix generated after fine-tuning BERT to a certain extent. But due to the low accuracy of fine-tuning BERT on emoji prediction task, it is inevitable to have a negative impact on the clustering results. And the pre-processing of data will

Table 2. Experimental results of baselines and TGCN-Bert

Model	Acc	Acc@5	Acc@C
FastText	36.3	72.4	47.9
DeepMoji	38.3	74.7	48.8
2-BiLSTM <sub>α</sub>	39.4	75.8	49.3
Utp-BiLSTM	39.2	74.7	48.4
BERT	39.7	76.3	<b>52.2</b>
BertGCN	37.9	<b>77.5</b>	49.1
TGCN-Bert	<b>42.5</b>	77.4	52.0

also greatly affect the model effect, while our pre-processing may be slightly simple and rough. If an emoji prediction information system with high accuracy is proposed and the correlation of emoji is extracted according to the confusion matrix generated by the model, the reliability and rationality of the clustering results can be further guaranteed. This can also be achieved by using emoji knowledge graphs to mine the connections between emoji. We would leave these in future work.

## **ACKNOWLEDGMENT**

The author would like to thank the editor and anonymous reviewers for their contributions towards improving the quality of this paper.

## **DATA AVAILABILITY**

The data used to support the findings of this study are included within the article.

## **CONFLICTS OF INTEREST**

The author declares that there is no competing interest for this work, and no funding was received.

## **FUNDING STATEMENT**

This research received no external funding.

## REFERENCES

- Al-Ayyoub, M., Rabab'ah, A., Jararweh, Y., Al-Kabi, M. N., & Gupta, B. B. (2018). Studying the controversy in online crowds' interactions. *Applied Soft Computing*, 66, 557–563. doi:10.1016/j.asoc.2017.03.022
- Almomani, A., Alauthman, M., Shatnawi, M. T., Alweshah, M., Alrosan, A., Alomoush, W., Gupta, B. B., Gupta, B. B., & Gupta, B. B. (2022). Phishing website detection with semantic features based on machine learning classifiers: A comparative study. *International Journal on Semantic Web and Information Systems*, 18(1), 1–24. doi:10.4018/IJSWIS.297032
- Alowibdi, J. S., Alshdadi, A. A., Daud, A., Dessouky, M. M., & Alhazmi, E. A. (2021). Coronavirus pandemic (COVID-19): Emotional toll analysis on Twitter. *International Journal on Semantic Web and Information Systems*, 17(2), 1–21. doi:10.4018/IJSWIS.2021040101
- Bai, S., He, H., Han, C., Yang, M., Yu, D., Bi, X., Gupta, B. B., Fan, W., & Panigrahi, P. K. (2023). Exploring thematic influences on theme park visitors' satisfaction: An empirical study on Disneyland China. *Journal of Consumer Behaviour*, cb.2157. doi:10.1002/cb.2157
- Bai, S., Kolter, J. Z., & Koltun, V. (2018). *An empirical evaluation of generic convolutional and recurrent networks for sequence modeling*. arXiv preprint arXiv:1803.01271.
- Barbieri, F., Anke, L. E., Camacho-Collados, J., Schockaert, S., & Saggion, H. (2018). Interpretable emoji prediction via label-wise attention LSTMs. *Proceedings of the 2018 conference on empirical methods in natural language processing*, 4766–4771. doi:10.18653/v1/D18-1508
- Barbieri, F., Ballesteros, M., & Saggion, H. (2017, April). Are Emojis Predictable? *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, 105–111.
- Barbieri, F., Camacho-Collados, J., Ronzano, F., Anke, L. E., Ballesteros, M., & Basile, V. et al.. (2018, June). Semeval 2018 task 2: Multilingual emoji prediction. *Proceedings of The 12th International Workshop on Semantic Evaluation*, 24–33. doi:10.18653/v1/S18-1003
- Barbosa, A., Bittencourt, I. I., Siqueira, S. W., Dermeval, D., & Cruz, N. J. (2022). A context-independent ontological linked data alignment approach to instance matching. *International Journal on Semantic Web and Information Systems*, 18(1), 1–29. doi:10.4018/IJSWIS.295977
- Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., & Malinowski, M. (2018). *Relational inductive biases, deep learning, and graph networks*. arXiv preprint arXiv:1806.01261.
- Baziotis, C., Nikolaos, A., Kolovou, A., Paraskevopoulos, G., Ellinas, N., & Potamianos, A. (2018, June). NTUA-SLP at SemEval-2018 Task 2: Predicting Emojis using RNNs with Context-aware Attention. *Proceedings of the 12th International Workshop on Semantic Evaluation*, 438–444. doi:10.18653/v1/S18-1069
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., & Amodei, D. et al.. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33, 1877–1901.
- Cai, H., Zheng, V. W., & Chang, K. C. C. (2018). A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE Transactions on Knowledge and Data Engineering*, 30(9), 1616–1637. doi:10.1109/TKDE.2018.2807452
- Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). *Bert: Pre-training of deep bidirectional transformers for language understanding*. arXiv preprint arXiv:1810.04805.
- Evans, V. (2017). The emoji code: The linguistics behind smiley faces and scardy cats. *Picador*, 256.
- Felbo, B., Mislove, A., Søgaard, A., Rahwan, I., & Lehmann, S. (2017, September). Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 1615–1625. doi:10.18653/v1/D17-1169
- Gaurav, A. (2022). A comprehensive survey on machine learning approaches for malware detection in IoT-based enterprise information system. *Enterprise Information Systems*, 1–25.

- Gu, J., Vo, N. D., & Jung, J. J. (2022). Contextual Word2Vec model for understanding chinese out of vocabularies on online social media. *International Journal on Semantic Web and Information Systems*, 18(1), 1–14. doi:10.4018/IJSWIS.309428
- Gupta, A., Bhatia, B., Chugh, D., Himabindu, G. S. S. N., Sethia, D., Agarwal, E., & Garg, S. et al. (2021, November). Context-aware emoji prediction using deep learning. In *International Conference on Artificial Intelligence and Speech Technology* (pp. 244-254). Springer International Publishing.
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770-778.
- He, Q., Wang, H., & Zhang, Y. (2020, November). Enhancing generalization in natural language inference by syntax. *Findings of the Association for Computational Linguistics. EMNLP, 2020*, 4973–4978.
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780. doi:10.1162/neco.1997.9.8.1735 PMID:9377276
- Huang, L., Ma, D., Li, S., Zhang, X., & Wang, H. (2019, November). Text Level Graph Neural Network for Text Classification. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3444-3450. doi:10.18653/v1/D19-1345
- Jain, A. K., Yadav, A., Kumar, M., García-Peñalvo, F. J., Chui, K. T., & Santaniello, D. (2022). A Cloud-Based Model for Driver Drowsiness Detection and Prediction Based on Facial Expressions and Activities. *International Journal of Cloud Applications and Computing*, 12(1), 1–17. doi:10.4018/IJCAC.312565
- Kim, Y. (2014). *Convolutional neural networks for sentence classification*. arXiv preprint arXiv: 1408.5882. 10.3115/v1/D14-1181
- Kipf, T. N., & Welling, M. (2016). *Semi-supervised classification with graph convolutional networks*. arXiv preprint arXiv:1609.02907.
- Kralj Novak, P., Smailović, J., Sluban, B., & Mozetič, I. (2015). Sentiment of Emojis. *PLoS One*, 10(12), e0144296. doi:10.1371/journal.pone.0144296 PMID:26641093
- Lin, Y., Meng, Y., Sun, X., Han, Q., Kuang, K., Li, J., & Wu, F. (2021). *Bertgcn: Transductive text classification by combining gcn and bert*. arXiv preprint arXiv:2105.05727. 10.18653/v1/2021.findings-acl.126
- Liu, R. W., Guo, Y., Lu, Y., Chui, K. T., & Gupta, B. B. (2022). Deep network-enabled haze visibility enhancement for visual IoT-driven intelligent transportation systems. *IEEE Transactions on Industrial Informatics*, 19(2), 1581–1591. doi:10.1109/TII.2022.3170594
- Liu, X., Luo, Z., & Huang, H. (2018). *Jointly multiple events extraction via attention-based graph information aggregation*. arXiv preprint arXiv:1809.09078. 10.18653/v1/D18-1156
- Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., & Stoyanov, V. (2019). *Roberta: A robustly optimized bert pretraining approach*. arXiv preprint arXiv:1907.11692.
- Lu, Z., Du, P., & Nie, J. Y. (2020). VGCN-BERT: augmenting BERT with graph embedding for text classification. *Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14–17, 2020 Proceedings, 42(Part I)*, 369–382.
- Lv, L., Wu, Z., Zhang, L., Gupta, B. B., & Tian, Z. (2022). An edge-AI based forecasting approach for improving smart microgrid efficiency. *IEEE Transactions on Industrial Informatics*, 18(11), 7946–7954. doi:10.1109/TII.2022.3163137
- Ma, W., Liu, R., & Wang, L. (2020). *Emoji prediction: Extensions and benchmarking*. arXiv preprint arXiv:2007.07389.
- Miller, H., Kluver, D., Thebault-Spieker, J., Terveen, L., & Hecht, B. (2017, May). Understanding emoji ambiguity in context: The role of text in emoji-related miscommunication. *Proceedings of the International AAAI Conference on Web and Social Media*, 11(1), 152-161. doi:10.1609/icwsm.v11i1.14901

Mintz, M., Bills, S., Snow, R., & Jurafsky, D. (2009, August). Distant supervision for relation extraction without labeled data. *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, 1003-1011. doi:10.3115/1690219.1690287

Na'aman, N., Provenza, H., & Montoya, O. (2017, July). Varying linguistic purposes of emoji in (Twitter) context. *Proceedings of ACL 2017, Student Research Workshop*, 136-141.

Singh, S. K., & Sachan, M. K. (2021). Classification of code-mixed bilingual phonetic text using sentiment analysis. *International Journal on Semantic Web and Information Systems*, 17(2), 59–78. doi:10.4018/IJSWIS.2021040104

Stergiou, C. L., Psannis, K. E., & Gupta, B. B. (2021). InFeMo: Flexible big data management through a federated cloud system. *ACM Transactions on Internet Technology*, 22(2), 1–22. doi:10.1145/3426972

Xu, Z., He, D., Vijayakumar, P., Gupta, B. B., & Shen, J. (2023). Certificateless public auditing scheme with data privacy and dynamics in group user model of cloud-assisted medical wsns. *IEEE Journal of Biomedical and Health Informatics*, 27(5), 2334–2344. doi:10.1109/JBHI.2021.3128775 PMID:34788225

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., & Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in Neural Information Processing Systems*, 32.

Yao, L., Mao, C., & Luo, Y. (2019, July). Graph convolutional networks for text classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(1), 7370–7377. doi:10.1609/aaai.v33i01.33017370

Ye, Z., Jiang, G., Liu, Y., Li, Z., & Yuan, J. (2020). Document and word representations generated by graph convolutional network and bert for short text classification. *ECAI 2020*, 2275-2281.

Yen, S., Moh, M., & Moh, T. S. (2021). Detecting compromised social network accounts using deep learning for behavior and text analyses. *International Journal of Cloud Applications and Computing*, 11(2), 97–109. doi:10.4018/IJCAC.2021040106

Zayats, V., & Ostendorf, M. (2018). Conversation Modeling on Reddit Using a Graph-Structured LSTM. *Transactions of the Association for Computational Linguistics*, 6, 121–132. doi:10.1162/tacl\_a\_00009

Zhang, X., Zhou, J., & Ji, D. (2020). Hierarchical Attention Emoji Prediction Model for Social Media. *Research on Computer Application*, 7, 1931–1934.

Zhangping Yang graduated from Chongqing High-tech Industrial Park in 2021. Worked in Hongqing High-tech Industrial Park. His research interests include prompt learning, text classification, sentiment analysis. Graduated from Hongqing High-tech Industrial Park in 2010. Worked in Hongqing High-tech Industrial Park. Her research interests include big data, database, data analysis. Graduated from Qingdao Technology University in 2020. Worked in Hongqing High-tech Industrial Park. His research interests include VQA, database, cross-modal generation.