entropy

MDPI

Article

BHGAttN: A Feature-Enhanced Hierarchical Graph Attention Network for Sentiment Analysis

Junjun Zhang [1], Zhengyan Cui [1], Hyun Jun Park [2] and Giseop Noh [2,*]

[1] Department of Computer Information Engineering, Cheongju University, Cheongju 28503, Republic of Korea
[2] Division of Software Convergence, Cheongju University, Cheongju 28503, Republic of Korea
* Correspondence: kafa46@cju.ac.kr

Abstract: Recently, with the rise of deep learning, text classification techniques have developed rapidly. However, the existing work usually takes the entire text as the modeling object and pays less attention to the hierarchical structure within the text, ignoring the internal connection between the upper and lower sentences. To address these issues, this paper proposes a Bert-based hierarchical graph attention network model (BHGAttN) based on a large-scale pretrained model and graph attention network to model the hierarchical relationship of texts. During modeling, the semantic features are enhanced by the output of the intermediate layer of BERT, and the multilevel hierarchical graph network corresponding to each layer of BERT is constructed by using the dependencies between the whole sentence and the subsentence. This model pays attention to the layer-by-layer semantic information and the hierarchical relationship within the text. The experimental results show that the BHGAttN model exhibits significant competitive advantages compared with the current state-of-the-art baseline models.

Keywords: sentiment analysis; BERT intermediate layer; hierarchical information encoding; hierarchical graph attention network

Citation: Zhang, J.; Cui, Z.; Park, H.J.; Noh, G. BHGAttN: A Feature-Enhanced Hierarchical Graph Attention Network for Sentiment Analysis. *Entropy* **2022**, *24*, 1691. https://doi.org/10.3390/e24111691

Academic Editor: Yuan Zong

Received: 18 October 2022
Accepted: 16 November 2022
Published: 18 November 2022

1. Introduction

The sentiment analysis task is one of the most classic tasks in NLP and plays a very important role in the field of NLP research. Early sentiment analysis methods mainly include traditional machine learning algorithms such as SVM [1], k-nearest neighbor, naive Bayes [2], etc. These methods are simple to implement and have high prediction accuracy and have achieved effective results in sentiment analysis tasks. However, these methods rely heavily on domain knowledge, and the text representation is high-latitude and sparse, and the feature expression ability is weak, which shows serious shortcomings in large-scale sample training. In recent years, the rapid development of deep learning has successfully promoted the research of sentiment analysis technology. The traditional learning algorithms relying on feature engineering have been completely changed by various end-to-end deep learning. The TextCNN model proposed by Kim [3] has obvious advantages in capturing local features. TextRNN [4] and its variant models [5,6] have short-term memory and can better express contextual information. However, these models cannot model longer sequence information and are not effective in dealing with long-range dependencies.

In recent years, with the emergence of transformers [7], large-scale pretraining models with attention mechanisms such as the core GPT [8–10], T5 [11], BERT [12], etc., have successively refreshed many NLP fields, and more and more researchers have begun to pay attention to the application of large-scale pre-training models. However, due to the complexity of natural language structure, the above methods usually model the entire text, and consider less the semantic structure inside the text. However, in the practice of sentiment analysis, there are many mixed emotions in many texts. For example, a sentence has a positive emotion, a neutral emotion, and a negative emotion. If the semantic modeling

of mixed-emotion sentences with multiple emotional tendencies is directly carried out on the whole, it may increase the difficulty of emotion judgment of the emotion analysis model, which makes it difficult for the model to be applied to the classification of mixed-emotion sentences. Intuitively, if the structural relationship of the sentence is considered, it will help the judgment of emotional polarity.

Recently, GNNs [13,14] have been shown to have strong representational capabilities in modeling structural information. The TextGCN model proposed by Yao et al. [15] builds a heterogeneous graph based on the relationship between documents and words, enabling the semisupervised classification of text on GCN. Huang et al. [16] improved the TextGCN model and proposed to generate a graph for each text, which saves memory while ensuring the effect is improved. Lin et al. [17] proposed to take advantage of large-scale pretraining models and graph networks to embed nodes in text graphs with BERT for initial word embedding, and then jointly train BERT and GCN modules to influence the representation of training data and unlabeled test data, achieving SOFT results on a wide range of text classification datasets. However, there are still some deficiencies in the current research on the construction graph method. For example, once the graph based on the global structure of TextGCN [15] is established, it cannot dynamically perceive the structural information inside a single document according to context semantics. Compared with the graph based on the global structure, TLGNN [16] can better learn the word-level relationship within a single text, but it does not pay attention to the dependencies between sentences within the document and cannot capture the structural information within the sample.

Based on the above analysis, we propose a novel modeling approach based on a large-scale pretrained model—BERT—and a hierarchical graph network. Different from the previous work, on the one hand, we use the language knowledge of the hidden state in the middle of BERT to enhance the semantic representation, and propose a BERT-based hierarchical information encoding method. This is different from the previous hierarchical coding model. For example, HAN [18] adopts the strategy of coding each level separately and then merging. This approach ignores the influence of the overall context when encoding the semantic information of subsentences. The pretrained language model BERT is able to pay attention to the correlation between the information at the subsentence level due to the task designed by NSP. At the same time, since BERT is a multilayer bidirectional encoder, the granularity of information extracted by BERT increases with the increase in the number of layers. By using each layer, more information can be fully introduced to enhance the semantic representation. On the other hand, in order to better learn the hierarchical relationship between sentences within the text, we consider that it is more reasonable to combine BERT's layers with the hierarchy of the text for modeling. The constructed model not only focuses on the semantic information layer by layer, but also the hierarchical relationship between sentences.

To the best of our knowledge, we are the first to propose a network model that combines the intermediate hidden layers of BERT and the structural layers of sentences to construct graphs. The contributions of this paper are as follows:

(1) A BERT-based hierarchical information encoding method is proposed. The average pooling layer is added to the original BERT layers to extract the semantic information of each subsentence at different layers. Since the encoding of the subsentence is derived from the overall encoding of BERT, the overall semantic information can be considered when encoding the hierarchical information.

(2) We propose a novel way of constructing graphs. Our method establishes a hierarchical graph structure based on the hierarchical relationship between BERT layers and sentences and use the graph attention network to extract the hierarchical structure features of the input text to build a multilevel hierarchical relationship graph (directed graph).

(3) We propose a novel sentiment analysis model, BHGAttN. The model aggregates semantic features from BERT and sentence structure features after graph training. It not only considers the semantic information, but also pays attention to the structural

information between sentences. As a result, BHGAttN can effectively improve the classification performance.

(4) We demonstrate that our method outperforms state-of-the-art baseline models through experiments on three datasets.

2. Related Work

2.1. Text Feature Representation

Sentiment analysis is an important application of text classification. The core problem that determines text classification is text representation, and text vectorization is an important method of text representation. Common methods of text vectorization include discrete representations, such as one-hot encoding, the bag-of-words model (BOW) [19], word frequency-inverse document frequency (TF-IDF) [20], n-gram [21], etc. The characteristics of this method are that the data are high-dimensional and sparse, and the computational complexity of the model is high. At the same time, because the lexical and word order are not considered, the relationship between word vectors cannot be measured, which makes it unable to fully represent different semantic information. The other is based on distributed representation, such as Word2vec [22,23], Glove [24], ELMO [25], GPT [8–10], etc. The distributed representation method is based on the language model technology, and the word vector is obtained through the training of the neural network. These methods overcome the limitation of dimensionality and improve the generalization ability of language models, and the obtained text representations can take into account the semantic environment of the context. Recently, based on the large-scale pretrained language model BERT, the model has strong scalability by fine-tuning transfer learning, and has achieved very good results in multiple NLP tasks. In the text, the granularity of extracting information for each layer of BERT is different. In order to introduce more information to enhance the semantic representation, we use the output of hidden states of each layer of BERT to initialize the embedded representation of graph nodes.

2.2. Graph Neural Networks

Graph neural networks extract and excavate the features and patterns of graph structure data through the mechanism of message passing. The existing research has proved the effectiveness of the graph-based text classification model. For example, the HR-DGCNN model proposed by Peng et al. [26] regards an article as a graph composed of word nodes and uses a convolutional network of semantic combination to realize topic classification. Zhang et al. [27] proposed to generate a text-level graph model TextING with global parameter sharing for each input text. Compared with the TextGCN [15] model, the TextING model eliminates the dependency burden between corpora, and its performance is better than the graph model built on the whole corpus. Recently, it has been found that the large-scale pretraining model is beneficial to tap the potential of graph learning. Yang et al. [28] proposed GraphFormers, a network architecture that deeply integrates GNN and PLM. This model adopts a hierarchical integration method of GNN and transformer block, enabling interactive training of text representation and graph aggregation. The experimental results show that the prediction accuracy of GraphFormers was greatly improved. Yang et al. [29] combined the advantages of BERT's semantic encoding and GCN's structural encoding and proposed a BEGNN model that considered both semantic and structural information and verified the effectiveness of the model on multiple datasets. These methods all build graphs for fine-grained word-level relationships in terms of composition. On the one hand, these models only focus on the short-distance semantic dependencies between words, ignoring the hierarchical relationship between sentences within the sample, which limits the expressiveness of graphs to a certain extent. On the other hand, the combination of GNNs and large-scale pretrained models is limited to shallow feature combinations without deep mining of BERT's inherent representational capabilities.

Inspired by [30], we consider that the granularity of semantic information extracted by each layer of BERT is different. In the practice of classification tasks, if only the last hidden

layer of the BERT model is used as the output, some information, such as phrase-level information, may be lost. Wait. Therefore, our model proposes to utilize information from the intermediate hidden layers of BERT for semantic modeling, while utilizing GAT [31] to map the hierarchical relationships between subsentences in the sample. In this way, the semantic information of sentences can be fully characterized, and the structural relationship between sentences can be captured. Intuitively, our modeling idea can better handle text classification tasks.

3. Method

In this section, we describe our modeling approach in detail. First, we show the overall architecture of the model. Second, we detail the specific methods of model implementation.

3.1. Model Architecture

The BHGAttN model we designed is shown in Figure 1. The model can be divided into three parts: (1) BERT-based hierarchical information encoding module; (2) GAT-based hierarchical graph network feature extraction; (3) feature fusion classification module. Firstly, the hierarchical relationship between the layers and subsentences of BERT is composed, and a hierarchical relationship graph (directed graph) is established, which considers both the semantics of layers and the hierarchical structure of sentences. The nodes in the graph are subsentence nodes and whole-sentence nodes of each layer of BERT. The features of the subsentence nodes are obtained by encoding the BERT information, and the feature representation of the whole-sentence nodes are obtained by random initialization (the left half of the figure). Nodes between different layers are also connected correspondingly. Secondly, the graph is constructed, it is put into the GAT model for training, and the whole-sentence node representation representing the structural feature is extracted (the right half of the graph). Then, the representations of the whole-sentence node at each level are fused through the attention mechanism to obtain the final hierarchical structure feature representation. Finally, we take out the representation of the first token position output from the last layer of BERT (that is, the overall text semantic representation) and fuse it with the hierarchical structure feature representation extracted through GAT. The implementation method of each module is introduced in following subsections.

3.2. BERT-Based Hierarchical Information Coding

Extract the semantic information of each subsentence at different layers. The method is to add a mean pooling layer to the original BERT layers. Figure 2 shows the method for encoding the hierarchical information of BERT.

Suppose a text S containing n subsentences is input, denoted as $S = \{s_1, s_2, \ldots, s_n\}$, where s_i represents the representation of the i-th subsentence. Each subsentence contains at most l words, then there is $s_i = \{w_{i,1}, w_{i,2}, \ldots, w_{i,l}\}$ for the i-th subsentence s_i. First, take S as a whole, and insert two special characters "[CLS]" and "[SEP]" at the beginning and end, respectively, to indicate the beginning and end of the sentence, so as to process it into an input format suitable for BERT.

$$\hat{S} = [[\text{CLS}], w_{1,1}, w_{1,2}, \ldots, w_{1,l}, w_{2,1}, w_{2,2}, \ldots, w_{2,l}, \ldots, w_{n,1}, w_{n,2}, \ldots, w_{n,l}, [\text{SEP}]] \quad (1)$$

$\hat{S}$ is then encoded using a BERT with L layers. For the j-th layer of BERT, the hidden layer representation H^{j} of $\hat{S}$ in this layer can be obtained:

$$\mathrm{H}^{\mathrm{j}} = \mathrm{BERT}^{\mathrm{j}}(\hat{S}) = \left[\mathrm{h}^{\mathrm{j}}_{[\mathrm{CLS}]}, \mathrm{h}^{\mathrm{j}}_{1,1}, \mathrm{h}^{\mathrm{j}}_{1,2}, \ldots, \mathrm{h}^{\mathrm{j}}_{\mathrm{n,l}}, \mathrm{h}^{\mathrm{j}}_{[\mathrm{SEP}]}\right] \in \mathbb{R}^{(\mathrm{nl}+2)\times \mathrm{d}} \quad (2)$$

where d represents the dimension of the hidden layer vector. In order to obtain the semantic representation of the i-th subsentence in the j-th layer, the mean pooling operation is applied

on the latent vector of the word to which the *i*-th subsentence belongs in the *j*-th layer (as shown by the red box in Figure 1):

$$\mathrm{h}_{\mathrm{i}}^{\mathrm{j}} = \mathrm{MeanPooling}\left(\left[\mathrm{h}_{\mathrm{i},1}^{\mathrm{j}}, \mathrm{h}_{\mathrm{i},2}^{\mathrm{j}}, \ldots, \mathrm{h}_{\mathrm{i},l}^{\mathrm{j}}\right]\right) \in \mathbb{R}^{1\times \mathrm{d}} \tag{3}$$

Furthermore, the hidden layer vector $\mathrm{h}_{[\mathrm{CLS}]}^{\mathrm{L}}$ at the "[CLS]" position of the last layer of BERT is usually used to represent the overall textual semantics.

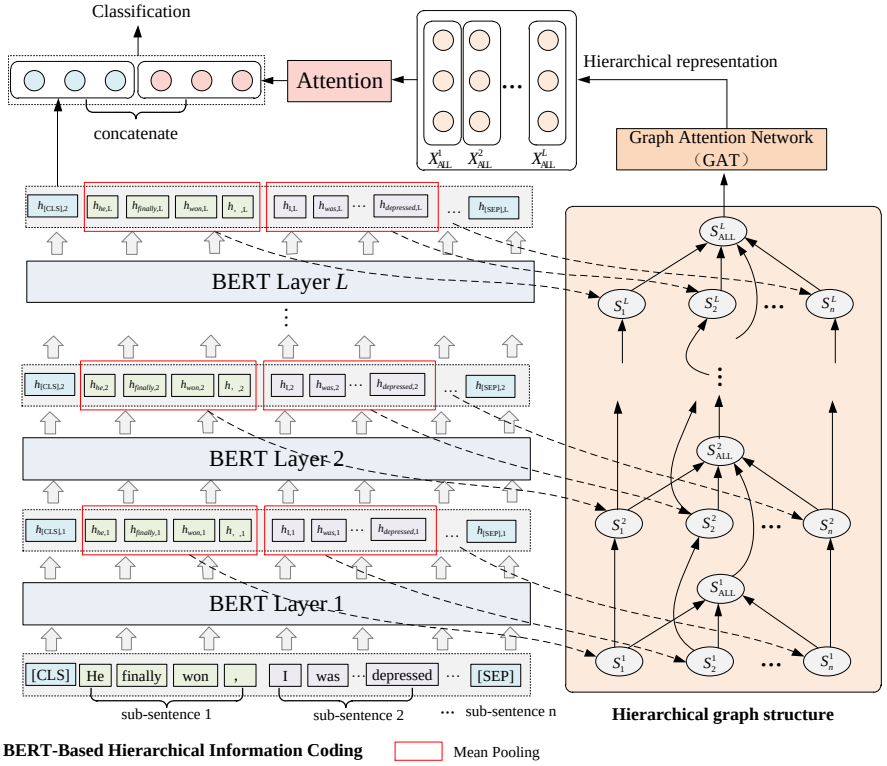

Figure 1. Architecture of BHGAN model. The left half of the figure is the BERT-based hierarchical encoding, and the right half is the graph construction and GAT-based feature extraction. The above is to use the representation of the overall nodes at each level through the attention mechanism to obtain the final hierarchical structure representation. The red border represents the mean pooling operation on the latent vector.

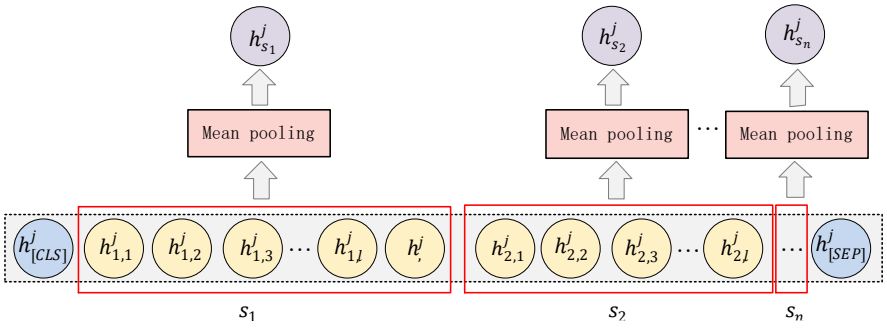

Figure 2. BERT-based information encoding.

3.3. GAT-based Hierarchical Graph Network

After obtaining the hierarchical information encoding of each subsentence in Equation (1), this paper constructs a hierarchical graph network $G = (V, E)$ as shown in the right half of Figure 1, where V is the set of nodes and E is the set of edges. The network is nested

by L layers, which correspond to each layer of BERT one by one. G contains two types of nodes: subsentence nodes (such as S_1^1) and whole-sentence nodes (such as S_{ALL}^1). For the same layer, a directed connection is made between the subsentence node and the whole-sentence node, such as $S_1^1 \rightarrow S_{ALL}^1$; between different layers, a directed connection is made between the node of the previous layer and the corresponding node of the next layer, such as $S_1^1 \rightarrow S_1^2, S_{ALL}^1 \rightarrow S_{ALL}^2$. Through graph G, the complete hierarchical structure of the sentence is constructed, and the adjacency matrix is shown in Figure 3.

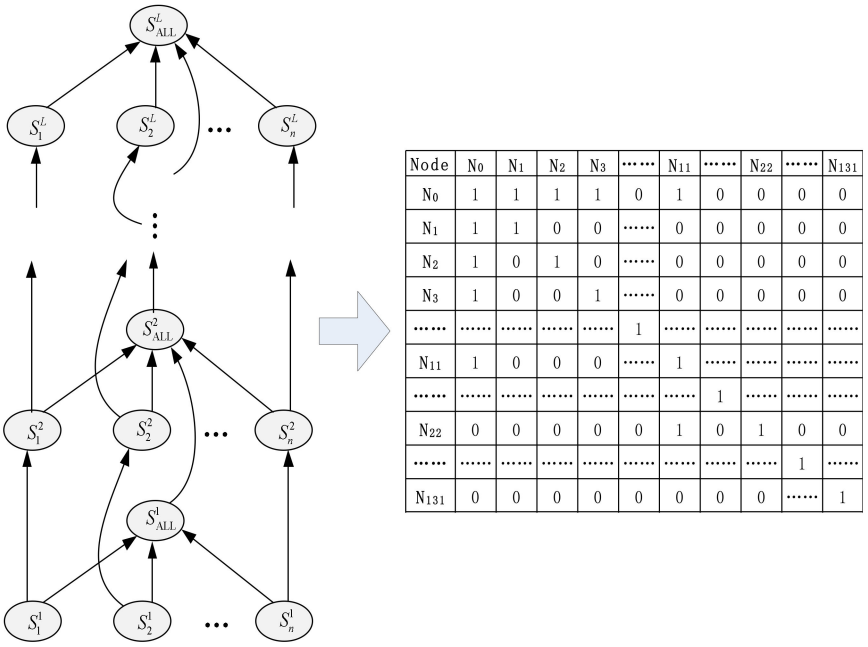

Figure 3. Adjacency matrix. For example, for a sample containing 7 subsentences, first expand the number of subsentences to the maximum number of subsentences (set to 10 in this paper) and pad in 0 if the number is not enough. If 12 layers of BERT are used, including whole-sentence nodes, there are $(10+1)*12 = 132$ nodes in total. Each node is represented by $\mathrm{N_i}(\mathrm{i} = 0, 1, 2, \cdots 131)$. Adjacency matrix $A \in \mathbb{R}^{|V|\times|V|}$, where V is the number of nodes.

After constructing the above-mentioned hierarchical graph G, it is necessary to assign the initial node feature representation h to all nodes. For the i-th subsentence node in the j-th layer, its initial node feature representation can be obtained by Equation (3). For the whole-sentence node, its initial feature representation is obtained by random initialization.

After obtaining the graph and the initialized representation h corresponding to the node, the graph attention network GAT is used to extract structural features from the graph. For a GAT with K layers and T heads, the calculation process can be described as follows:

(1) First, GAT needs to calculate the attention weight of each node in the node i and its connected node set in each layer:

$$\alpha_{ij}^t = \frac{\exp\left(\mathrm{LeakyReLU}\left(\mathrm{W}_3^t\left[\mathrm{W}_1^t x_i^{(k)} \| \mathrm{W}_2^t x_j^{(k)}\right]\right)\right)}{\sum_{q\in\mathcal{N}_i}\exp\left(\mathrm{LeakyReLU}\left(\mathrm{W}_3^t\left[\mathrm{W}_1^t x_i^{(k)} \| \mathrm{W}_2^t x_q^{(k)}\right]\right)\right)} \tag{4}$$

In the above formula, k represents the k-th layer of GAT; t represents the t-th attention head; $\mathrm{W}_1^t, \mathrm{W}_2^t, \mathrm{W}_3^t$ are all learnable weight matrices; $||$ represents the concatenation operation.

(2) After the attention weights are obtained, the representation of node i in the next layer can be updated by the weighted summation of the neighbor nodes:

$$h_i^{(k+1)} = \Big\Vert_{t=1}^{T} \tanh\left(\sum_{j\in\mathcal{N}_i} \alpha_{ij}^t W_4^t h_j^{(k)}\right) \tag{5}$$

where W_4^t is the learnable weight matrix. After the structural features are extracted by GAT, the representations of all the whole-sentence nodes in the last layer $X_{ALL} = \left[X_{ALL}^1, X_{ALL}^2, \ldots, X_{ALL}^L\right]$ are fused through the attention mechanism to obtain the final hierarchical structure feature representation:

$$\beta = \text{softmax}\left(W_\beta X_{ALL} + b_\beta\right) \tag{6}$$

$$H_{\text{Structure}} = \sum_{i=1}^{L}\left(\beta_i X_{\text{ALL}}^i\right) \tag{7}$$

where β is the attention weight, W_β is the learnable weight matrix, and b_β is the bias.

3.4. Fusion Classification Module

After obtaining the overall text semantic representation in Equation (1) and the hierarchical structure representation in Equation (2), the two representations are connected and reduced to the classification dimension through a linear layer, and Softmax is used to predict its category:

$$p(y) = softmax\left(W_y F + b_y\right) \tag{8}$$

where $F = x_{[\text{CLS}]}^L \| X_{\text{Structure}}$. W_y is a weight matrix that can be learned. b_y is biased. The loss function of model training is cross entropy loss function.

$$loss_c = -\sum_{i=1}^{N} y^i \log p\left(y^i\right) \tag{9}$$

where y^i represents the true class label of the i-th sample.

4. Experiment

In this section, we evaluate the effectiveness of the BHGAttN model with extensive experiments on 3 datasets.

4.1. Dataset

In our model, since BERT is a multilingual model, it can support multiple languages. In this paper, we test the performance of the model with Chinese, Korean, and English data as representatives. Among them, the English dataset is MR, a movie review dataset commonly used in sentiment analysis. The Chinese and Korean dataset were derived from online comments made by Chinese and Korean netizens on the 2021 Tokyo Olympic Games against the background of COVID-19 via crawlers. For the crawled data, we carried out data cleaning and manual labeling, and randomly sampled 30,000 labeled data according to the ratio of positive and negative labels as the training data of the model. Details of the dataset as shown in Table 1 Statistics of the dataset, including statistics of positive and negative sentiment polarity.

Table 1. Statistics of the dataset, including statistics of positive and negative sentiment polarity.

Dataset	Positive	Negative	Total
Ch_TOR (Chinese Olympic Review)	15,000	15,000	30,000
Ko_TOR (Korean Olympic Review)	5571	24,429	30,000
MR (Movie Review)	3554	7108	10,662

For the above datasets, we randomly split them in a ratio of 8:1:1 for model training, validation, and testing, respectively.

4.2. Baseline

To have a clearer comparison, we divide the baseline models into 3 groups for comparative experiments. Details are as follows:

Group 1: Traditional deep learning models based on BERT word embeddings.

In the experiments, we explore initializing the word vectors of the baseline model with BERT to obtain better embedding representation.

- BERT-TextCnn [3]: TextCNN is good at short-text feature extraction and is suitable for short-text comment sentiment classification.
- BERT [12]: An excellent baseline model that performs well on multiple NLP tasks.
- BERT-BiGRU [32]: In sentiment classification task, bidirectional GRU is used to extract features.
- BERT-BiGRU_Att [33]: Bidirectional gated recurrent unit (BiGRU) combined with attention mechanism is used to efficiently capture sequence context features.

Group 2: Graph-based text classification models.

This group of experiments mainly compares the effect of different ways of constructing the map on the results.

- TextGCN [15]: Huge single-text heterogeneous graph composed of word nodes and document nodes. Transforms a text classification problem into a node classification problem.
- TextING [27]: Constructs a vocabulary for each document. Whether there is an edge between two words is judged by sliding window method, and all samples are constructed into nodes at a time.
- BEGNN [29]: Constructs graphs on each text according to word co-occurrence relations, and fuses semantic features acquired by BERT and structural features captured by GCN through co-attention to obtain more effective representations.

Group 3: Ablation experiments.

In this set of experiments, in order to verify the effectiveness of each module of the model, we observe the impact on the results by removing or replacing submodules in the model.

- w/o_HGAN: The hierarchical graph network module in the model is removed, the feature representation of each subsentence of the last hidden layer is calculated and connected with the [CLS] token representation after fusion through the attention mechanism, and then classification prediction is performed.
- w/o_HBERT: The layered coding part based on BERT is removed, only the output of the last hidden layer of BERT is used to encode the graph nodes, and the experimental results are observed.
- BERT_HGCN: In order to verify the influence of GAT and GCN on the experimental results, we replace the GAT module with GCN for training.
- RoBERTa_HGAT: We replace the BERT module with Roberta for training and observe the experimental results.

4.3. Experimental Setup

In the experiment, the value of the hyperparameter is mainly set according to the previous work experience. In the model architecture, we use BERT-Base as the pretraining language model, which can be well migrated to other transformer based pretraining language models. The dimension of BERT Base hidden layer is 768. The attention of the GAT network is consistent with the number of attention heads in BERT, set to 12, and the dimension of each head is 64. The learning rate is set to 1e-5 and the optimizer is Adam. Dropout is set to 0.5 after the fully connected layer. The epoch of the training is set to 100. For the determination of the maximum number of subsentences, it is set according to the average number of subsentences in different datasets. In addition, in order to better prevent overfitting, we use the early stop method in training. The maximum tolerance for improvement of the f1 value of the validation set in the model is set to 10; that is, the

training process is terminated when the performance of the model on the validation set (f1 score) does not improve in 10 consecutive iterations. For the baseline model, we use the same parameter settings as our model, which allows for a fair comparison with our model.

After setting the model parameters, the training, verification, and testing process will be automatically output. If each verification set is improved, a test will be run and the test results will be output. Finally, the performance of the model is subject to the accuracy of the final test set.

4.4. Experimental Results and Analysis

In the experimental results (Table 2), we use the accuracy on the test set as an evaluation metric for model performance. From Table 2, we can see that the performance of the graph-based model is overall higher than that of the traditional deep learning model. Our method (the third group) performed the best.

Table 2. Test accuracy (%) of each model on 3 datasets.

Category	Model	Dataset		
		Ch_TOR	Ko_TOR	MR
BERT-based traditional model	BERT_TextCNN	77.60	86.56	77.69
	BERT_BiGRU	76.13	84.86	76.1
	BERT_BiGRU-Att	76.67	86.12	77.32
	BERT	81.83	86.62	86.22
Graph-based model	TextGCN	-	-	76.74
	TextING	-	-	78.74
	BEGNN	-	-	84.47
Ablation experiment (Ours)	**BHGAttN**	**82.63**	**87.79**	**87.72**
	w/o_HBERT	81.96	86.63	86.68
	w/o_HGAT	82.26	87.06	86.22
	BERT_HGCN	82.03	87.29	86.32
	RoBERTa_HGAT	**83.27**	**-**	**88.85**

The TextCNN model in the first set of experiments achieves the best performance except for the BERT model with better word embeddings. This is inseparable from its ability to effectively model the semantics of continuous short texts. Similarly, the sequence model BiGRU with pretrained word embeddings also has excellent performance, and the performance of the BiGRU-ATT model of BIRGU with the addition of the attention mechanism has been significantly improved. Compared with traditional deep learning models, the BERT model still maintains the most competitive results. This shows the absolute advantage of large-scale pretrained language models in semantic modeling.

The second group of graph-based methods outperformed the first group overall, indicating that graph networks are effective for text processing. The TextING model improves the composition of the TextGCN model, so that each document has its own graph structure, and the structural information inside the document is well mined. Therefore, its performance is better than the TextGCN model. The BEGNN model, which combines the advantages of the large-scale pretraining model and the TextING model, has achieved the most satisfactory results. This proves that large-scale pretrained models are beneficial to tap the potential of graph learning.

The third group of experiments is our method. Table 2 shows that the BHGAttN model achieves the best results on all datasets. By comparing with the best baseline model, BEGNN, we notice that although the BEGNN model provides the feature interaction module of BERT and GNN, it has a great improvement in performance over other graph models. But our model takes full advantage of the encoded information of the intermediate layers of BERT and introduces a more adequate semantic representation. At the same time, we pay attention to the hierarchical structure between sentences, which can better reflect the structural dependency between document contents than the short-distance word

co-occurrence relationship. Experimental results on the MR dataset show that with the same parameter settings, our model outperforms BEGNN by 3.25%, which proves that our method is very effective.

4.4.1. Effectiveness of BERT-Based Hierarchical Information Coding

To examine the impact of different modules in the model on the overall performance of the model, In the experiment, we try to remove the BERT-based hierarchical information encoding module and name it the w/o_HBERT model. We fix the various components of the original model and keep the basic composition of BHGAttN. Only the average pooling layer is added to the output of the last layer of BERT to extract the semantic encoding of each subsentence as the initial representation of the subsentence nodes. The whole-sentence node is obtained by random initialization. After GAT learning, since the number of whole-sentence nodes is 1, we remove the attention feature fusion module, and the model architecture is shown in Figure 4.

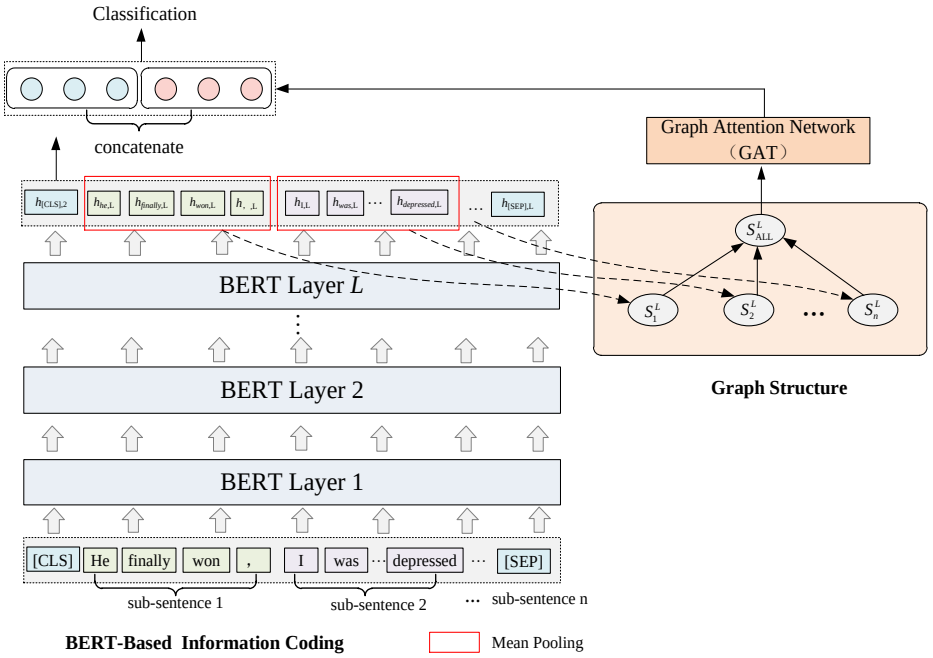

Figure 4. w/o_HBERT model architecture.

We observe from experiments that removing the layered BERT encoding will have a certain negative impact on the results, but our composition method is still better than the method of constructing text graphs based on word co-occurrence relationships. At the same time, it is also illustrated that using the information of the intermediate hidden state of BERT can enhance the semantic features, so that the graph network can obtain better initial embedding representation.

4.4.2. Effectiveness of Hierarchical Graph Attention Networks

We also investigate the impact of hierarchical graph attention networks on model performance. The specific method is to remove the right half of the model architecture (that is, the GAT network module), and name the model after removing the GAT network module w/o_HGAT (Figure 5). Similarly, we use the output of the last hidden state of BERT and obtain the representation of each subsentence through mean pooling, and then use the attention mechanism to fuse the representation of each subsentence. Finally, it is concatenated with the output of the [CLS] token of the last layer of BERT and mapped to the classification dimension through a linear layer for classification.

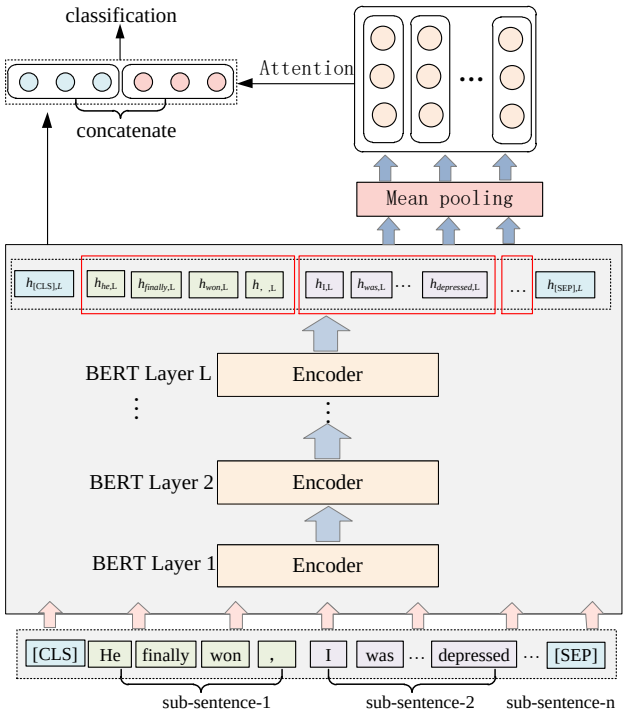

Figure 5. w/o_HGAT model architecture.

During the experiment, we noticed that the performance of the model was degraded by removing the GAT layered network module, indicating the effectiveness of the layered graph network in the model. However, at the same time, the experimental results show that w/o_HGAT is still higher than the [CLS] classification performance of the BERT model. It is proved that the subsentence feature representation can promote the performance of the model.

4.4.3. Influence of Different Graph Networks on Experimental Results

We also compare the effect of layered GAT composition and layered GCN composition on model performance. Replace the GAT network on the right part of the model architecture with GCN. The experimental results in Table 2 show that GCN-based models perform slightly lower than GAT. The possible reason is that when the GCN model learns the node representation, the weights of the edges are fixed, which limits the expressive ability of the edges to a certain extent. GAT, on the other hand, adaptively learns the edge weights through the attention mechanism, which makes it more effective at fusing the information of node features and graph structure.

4.4.4. Influence of Large-Scale Pretraining Model on Experimental Results

In order to further verify the advantages brought by the large-scale pretrained language model, we replaced the BERT pretrained language model with RoBERTa [34] for experiments. From the experimental results in Table 2, it can be seen that the performance of RoBERTa-HGAN is much improved than that of BHGAttN, which is because RoBERTa improves the pretraining method of BERT, which makes RoBERTa outperform BERT. It can be seen that an excellent large-scale pretraining model is beneficial to the model.

5. Conclusions

In this paper, in view of the problems existing in the current text classification task, we propose to make full use of the advantages of large-scale pretraining models and graph neural networks and design a text classification model based on BERT and GAT to model hierarchical relationships. Different from previous work, on the one hand, we propose a BERT-based hierarchical information encoding method to enhance semantic

features through the output of the intermediate hidden state of BERT; On the other hand, a hierarchical graph network corresponding to each layer of BERT is constructed, and the graph attention network is used to extract the hierarchical structure features of the input text. The model we constructed considers both the semantic features of layer-by-layer text and the hierarchical relationship between text contents. The experimental results demonstrate the effectiveness of the model. However, due to the huge number of parameters in the large-scale pretraining model, the method of constructing the graph layer-by-layer will increase the number of edges in the graph, which increases the burden of memory.

Therefore, in future work, we will explore how to further improve the performance of the model with low memory consumption. For example, we consider using a lightweight pretraining language model to replace BERT. Greatly reducing the number of parameters can reduce the calculation cost and greatly improve the training speed of the model. In addition, the cause of memory overload is closely related to the graph size. In the next step, we will dig deeper into the linguistic information encoded in the neural network model. Specifically, it is to explore the different performance of hidden layers in the middle and analyze which layers have more positive impact on semantic encoding. When modeling, we will consider removing less influential layers and reducing the size of the graph to some extent to reduce memory consumption.

Author Contributions: Methodology, J.Z.; analysis, J.Z. and G.N.; software, J.Z. and Z.C.; validation, J.Z., H.J.P. and G.N.; writing, J.Z.; review and editing, G.N. and H.J.P.; visualization, J.Z. and Z.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Key Research Projects Program of Higher Education Institutions in Henan Province, China. Grant number 22B520040.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data provided in this study is collected with crawler programs from the online comments in TOG news from Naver News Network (www.news.naver.com) in South Korea, Sina News Network (https://news.sina.com.cn) in China, and New York Times (https://www.nytimes.com/) in the United States. The download URL of the public dataset MR is https://github.com/CRIPAC-DIG/TextING (accessed on 15 November 2022).

Conflicts of Interest: The authors declare no conflict of interest regarding the publication of this paper.

References

1. Chen, P.H.; Lin, C.J.; Schölkopf, B. A tutorial on ν-support vector machines. *Appl. Stoch. Model. Bus. Ind.* **2005**, *21*, 111–136. [CrossRef]
2. Masurel, P. Of Bayesian Average and Star Ratings. Available online: https://fulmicoton.com/posts/bayesian_rating/ (accessed on 16 December 2021).
3. Kim, Y. Convolutional Neural Networks for Sentence Classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1746–1751.
4. Liu, P.; Qiu, X.; Huang, X. Recurrent Neural Network for Text Classification with Multi-Task Learning. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), New York, NY, USA, 9–15 July 2016.
5. Huang, Z.; Wei, X.; Kai, Y. Bidirectional LSTM-CRF Models for Sequence Tagging. *arXiv* **2015**, arXiv:1508.01991.
6. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv* **2014**, arXiv:1406.1078.
7. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All You Need. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Long Beach, CA, USA, 3–9 December 2017; pp. 5998–6008.
8. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. Improving Language Understanding by Generative pre-Training. 2018. Available online: https://www.cs.ubc.ca/~{}amuham01/LING530/papers/radford2018improving.pdf (accessed on 15 November 2022).
9. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners. OpenAI blog. *OpenAI Blog* **2019**, *1*, 9.
10. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Amodei, D. Language Models are Few-Shot Learners. In Proceedings of the Advances in Neural Information Processing Systems 33 (2020), Virtual, 6–12 December 2020; pp. 1877–1901.

11. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* **2020**, *21*, 1–67.
12. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
13. Grover, A.; Leskovec, J. node2vec: Scalable Feature Learning for Networks. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 855–864.
14. Dong, Y.; Chawla, N.V.; Swami, A. metapath2vec: Scalable Representation Learning for Heterogeneous Networks. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, 13–17 August 2017; pp. 135–144.
15. Yao, L.; Mao, C.; Luo, Y. Graph Convolutional Networks for Text Classification. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019.
16. Huang, L.; Ma, D.; Li, S.; Zhang, X.; Wang, H. Text level graph neural network for text classification. *arXiv* **2019**, arXiv:1910.02356.
17. Lin, Y.; Meng, Y.; Sun, X.; Han, Q.; Kuang, K.; Li, J.; Wu, F. BertGCN: Transudative Text Classification by Combining GCN and BERT. In Proceedings of the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021), Bangkok, Thailand, 1–6 August 2021; pp. 1456–1462.
18. Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A.; Hovy, E. Hierarchical Attention Networks for Document Classification. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human language Technologies, San Diego, CA, USA, 12–17 June 2016; pp. 1480–1489.
19. Joachims, T. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In Proceedings of the European Conference on Machine Learning, Chemnitz, Germany, 21–23 April 1998; Springer: Berlin/Heidelberg, Germany, 1998; pp. 137–142.
20. Ramos, J. Using tf-idf to Determine Word Relevance in Document Queries. In Proceedings of the First Instructional Conference on Machine Learning, Washington DC, USA, 21–24 August 2003; Volume 242.
21. Cavnar, W.B.; Trenkle, J.M. N-Gram-Based Text Categorization. In Proceedings of the SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval, Las Vegas, NV, USA, April 1994; Volume 161175.
22. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
23. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed Representations of Words and Phrases and Their Compositionality. In Proceedings of the Advances in Neural Information Processing Systems 26, Lake Tahoe, NV, USA, 5–10 December 2013; pp. 3136–3144.
24. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
25. Peters, M.; Neumann, M.; Iyyer, M.; Gardner, M.; Zettlemoyer, L. Deep Contextualized Word Representations. In Proceedings of the NAACL 2018, New Orleans, LA, USA, 1–6 June 2018; pp. 2227–2237.
26. Peng, H.; Li, J.; He, Y.; Liu, Y.; Bao, M.; Wang, L.; Yang, Q. Large-Scale Hierarchical Text Classification with Recursively Regularized Deep Graph-Cnn. In Proceedings of the 2018 World Wide Web Conference, Lyon, France, 23–27 April 2018; pp. 1063–1072.
27. Zhang, Y.; Yu, X.; Cui, Z.; Wu, S.; Wen, Z.; Wang, L. Every Document Owns Its Structure: Inductive Text Classification via Graph Neural Networks. In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 2020), Seattle, WA, USA, 5–10 July 2020; pp. 334–339.
28. Yang, J.; Liu, Z.; Xiao, S.; Li, C.; Lian, D.; Agrawal, S. GraphFormers: GNN-nested Transformers for Representation Learning on Textual Graph. In Proceedings of the Advances in Neural Information Processing Systems 34 (NIPS 2021), Online, 6–14 December 2021; pp. 28798–28810.
29. Yang, Y.; Cui, X. Bert-enhanced text graph neural network for classification. *Entropy* **2021**, *23*, 1536. [CrossRef] [PubMed]
30. Xiao, Z.; Wu, J.; Chen, Q.; Deng, C. BERT4GCN: Using BERT Intermediate Layers to Augment GCN for Aspect-based Sentiment Classification. *arXiv* **2021**, arXiv:2110.00171.
31. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y. Graph Attention Networks. In Proceedings of the International Conference on Learning Representations (ICLR), Vancouver, Canada, 30 April–3 May 2018; pp. 1–12.
32. Feng, X.; Liu, X. Sentiment Classification of Reviews Based on BiGRU Neural Network and Fine-Grained Attention. In Proceedings of the Journal of Physics: Conference Series, 2019 3rd International Conference on Machine Vision and Information Technology (CMVIT 2019), Guangzhou, China, 22–24 February 2019; IOP Publishing: Bristol, UK, 2019; Volume 1229, p. 012064.
33. Zhou, L.; Bian, X. Improved text sentiment classification method based on BiGRU-Attention. *J. Phys. Conf. Ser.* **2019**, *1345*, 032097. [CrossRef]
34. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arxiv* **2019**, arXiv:1907.11692.