2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

LBERT-MSDF: A Dynamic Fusion Network for Multi-Scale Text Feature Extraction

MingLe Zhou[1,2,3] , XinYu Liu[1,2] , JiaChen Li[1,2] and DeLong Han[1,2,*]

[1] Key Laboratory of Computing Power Network and Information Security, Ministry of Education, Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences), Jinan, China.
[2] Shandong Provincial Key Laboratory of Computing Power Internet and Service Computing, Shandong Fundamental Research Center for Computer Science, Jinan, China.
[3] SHANDONG SCICOM Information and Economy Research Institute Co.， Ltd.

Abstract. Text categorization is a core task in natural language processing and plays an irreplaceable role in tasks such as spam detection, user sentiment analysis, and news topic classification. In this paper, we propose a dynamic fusion network for multi-scale semantic feature extraction based on BERT in order to improve the model's understanding of the semantics, which is difficult to capture different levels of semantic information in the text at the same time, which leads to classification ambiguity and reduces the classification accuracy, etc. This network proposes a multi-functional channel depth modulator (VCDM) in order to improve the model's understanding of the semantics, which enables the model to perform a deeper analysis at the macro level such as the context of the text. The small-scale feature extractor (BMHA) is also proposed to perform feature extraction at the micro level to build the underlying structure of the vocabulary, which complements and enhances the effect of the VCDM module. In order to be able to capture the feature information better and optimize the accuracy of the classification results, the optimal BERT output level is selected for the input and output of the model, as well as the introduction of an adaptive weight weighting mechanism to dynamically fuse the outputs of the two modules. The experimental results show that the MSDF model outperforms the existing models, achieving better accuracies of 95.03\%, 93.63\%, and 89.47\% on the three datasets, respectively, proving the effectiveness of the MSDF model proposed in this paper on the text classification task.

Keywords: Natural Language Processing, Text Classification, Bert, Multi-scale feature Extraction , Weighted Fusion Strategy.

1 Introduction

1.1 A Subsection Sample

With the advent of the digital era, semantic feature extraction technology has a wide range of applications in various scenarios. For example, it helps enterprises extract valuable information from text data to optimize business processes, improve customer satisfaction and enhance decision support. And it plays a central role in several applications such as understanding user generation, content recommendation systems and social media.

The bi-directional training strategy of BERT [1] opens up new ways for text context understanding, but BERT still has limitations in capturing fine-grained morphemic information. To overcome this limitation, researchers have made many innovations, such as[2] combining CNN with BERT.

However, the existing methods often lead to classification ambiguity and lower classification accuracy, which limits the further improvement of model performance. To address this challenge, this paper proposes an innovative dynamic fusion network for multi-scale semantic feature extraction based on BERT.The innovations of the LBERT-MSDF model are:

Multi-functional Channel Depth Modulator (VCDM). specially optimized for the semantic and contextual level of the text.VCDM significantly improves the model's ability to capture and express semantic information in long text by fine-tuning the feature channels.

Small scale feature extractor (BMHA). Focusing on the analysis of the morpheme level, builds the underlying structure of the vocabulary, provides a complement to the VCDM module, and the two complement each other to enhance the model's comprehensive grasp of the text information.

Optimal input fusion level and output adaptive weighting strategy. The optimal BERT fusion level is determined through systematic experiments as the input of VCDM and BMHA. The adaptive weighting strategy is introduced, which allows the model to automatically adjust the proportion of feature fusion weights based on the training data and dynamically optimize the feature combination, thus significantly improving the accuracy of the classification task and the generalization ability of the model.

2 Related Work

2.1 Single Network Structure

In the field of natural language processing, the core challenge of the text categorization task is how to effectively capture and fuse multi-scale features in text.In recent years, with the development of deep learning techniques, researchers have proposed a variety of models to address this problem. Early research fo-cused on using a single network structure, such as CNN[3] and RNN[4]. Textcnn model [5], proposed in 2014, applies cnn to the text classification task, followed by Dpcnn [6], which utilizes multilayer convolutional layers to process the data. Text Recurrent Neural Network (TextRNN) [7] was proposed by Liu, Qiu and Huang, TextRNN is good at capturing long-distance dependencies in the text but may not be as effective as TextCNN in processing local features; Wang et al.proposed a memory network classification based on attention for long-term and short-term aspect-oriented emotions and a classification emotion based on a long-short-term memory (LSTM) network [8]. Cao et al. Brueckner and

Schulter built a BiLSTM model [9]that enables the model to take contextual information into account. proposed a bi-directional gated recurrent unit (BiGRU) [10] based innovation in its ability to process both forward and reverse text sequence infor-mation, thus capturing the contextual environment more comprehensively and providing a richer semantic representation.

2.2 Hypergraph Structural Learning

With the rise of pre-trained models, BERT has become the model of choice for many NLP tasks, and its powerful text representation has achieved significant performance gains in several benchmark tests. On this basis, researchers have begun to try to combine BERT with other neural network models to further enhance text classification. Zhao et al. established a classification model for dietary texts and utilized Word2Vec and LSTM to classify dietary-appropriate texts and taboo texts [11]. Zhou et al. proposed a multi-channel CNN and bidirectional gated recurrent unit (BiGRU) model based on the attention mechanism[12]. The model is not only able to focus on the words in the sentence that are important for sentiment polarity classification through the attention mechanism but also combines the advantages of CNN in extracting local features of the text and the BiGRU network in extracting contextual semantic information of long texts, which improves the model's textual feature extraction capability.Sentiment evolution analysis of microblog public opinion is studied, and the sentiment classification model is constructed by Word2Vec and SVM methods, and the Word2Vec word similarity computation model can effectively reflect the netizens' sentiment attitudes towards five types of subject objects and the thematic features within the stage of that public opinion[13]. As well as for the existing recurrent neural network models that have the problems of long training time and difficulty in obtaining target-related information, a multi-head attention-gated convolutional network with positinal embedding (PE-MAGCN) is proposed [14].

3 Preliminaries

3.1 overview

To address the problems of insufficient ambiguity processing and inadequate contextual understanding currently faced in text processing, this paper proposes a parallel classification model, MSDF, which integrates the output of BERT from the semantic and semantic dimensions. The model structure of MSDF is shown in Fig. 1, and the design concept lies in the deep mining of the semantic and semantic level features of the textual data through parallel processing.

First, the model uses the BERT pre-training model to convert the textual data into a high-dimensional vector representation. The (last_hidden_state) attribute of BERT provides context-sensitive token representations, where the representation of the [CLS] token at the beginning of the sequence is used as an input for semantic features into the module VCDM. The rest of the token vectors are spliced as input for the semantic features into the module BMFA.

Next, these two sets of inputs are processed independently by the multifunctional channel depth modulator VCDM and the small-scale feature extractor BMHA, respectively. The VCDM combines the processing power of the adaptive feature modulator AFM and the deep convolutional unit to enhance the performance of representing and categorizing long textual semantic information. The AFM enhances the numerical stability by dynamically adapting the multidimensional representation of the input features and enhances the model performance through the learnable activation function and gating mechanism. Learning activation function and gating mechanism to enhance the model performance. Deep convolution and pooling operations are then used for feature extraction, and the fully connected layer is used for classification prediction. On the other hand, the BMHA module utilizes the multi-head attention mechanism to learn multiple representations of the input data in parallel to extract richer morphemic features. The features processed by the attention mechanism are fed into the bi-directional GRU layer for a deeper understanding of the contextual dependencies in the sequence data. The processed semantic and morphemic feature classification results are integrated into the dynamic fusion stage. We experimentally determine the optimal initial weight parameters and update these weights by gradient descent algorithm during the training process so that the model can automatically adjust the fusion ratio of different outputs to achieve the best classification results. With this strategy of parallel processing and dynamic fusion, the MSDF model proposed in this paper is not only able to comprehensively capture the semantic and morphemic features of the text but also able to achieve significant performance improvement in the classification task. Figure 1 shows the general architecture of MSDF and the specific implementation details of VCDM and BMHA.The input data is processed by two independent modules: the Versatile Channel Depth Modulator (VCDM) and the Small Scale Feature Extractor (BMHA).The VCDM integrates an Adaptive Feature Modulator (AFM) and a deep convolutional unit designed to enhance the semantic representation and classification performance of long texts.

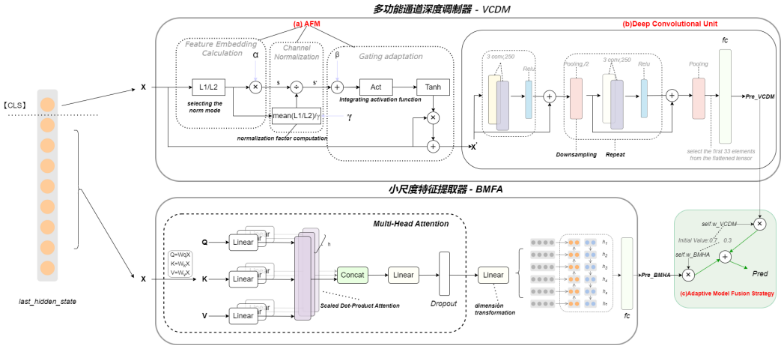

Fig. 1. The input data is processed by two independent modules: the Versatile Channel Depth Modulator (VCDM) and the Small Scale Feature Extractor (BMHA).The VCDM integrates an Adaptive Feature Modulator (AFM) and a deep convolutional unit designed to enhance the semantic representation and classification performance of long texts.

3.2 Bert and level selection

In our comprehensive review of the BERT output, we paid particular attention to two key output components: the last layer [CLS] marker and the hidden state of the last layer. The [CLS] marker, as the starting symbol of the BERT sequence, is strategically placed at the top of the input sequence. The vector representation of this marker is designed to aggregate information from the entire input sequence, and so plays a crucial role in capturing the overall semantics. In our model, the output of [CLS] tokens is directly extracted and used as input to the semantic processing module VCDM, which provides a robust representation of sentence-level features for downstream tasks.

Further, the output of the last layer of BERT was carefully analyzed. Except for [CLS] tokens, the rest of the token vectors of this layer are carefully extracted and spliced to serve as inputs to the BMHA module, a small-scale feature extractor. This splicing operation not only preserves the local syntactic and semantic information but also further enhances the model's comprehensive grasp of the textual content by combining this information with the global semantic representation of the [CLS] tokens.

4 Methodology

4.1 Versatile Channel Depth Modulator.

In this section, we will detail the multifunctional channel depth modulator VCDM proposed in this paper. By combining the dynamic feature modulation capability of the adaptive feature modulator AFM (As shown Fig.1-a) with the processing of the deep convolutional unit (As shown Fig.1-b) in order to enhance the performance of the task of representing and categorizing the semantic information of long texts. Figure 1 illustrates the general architecture of our proposed VCDM. We first obtain the [cls] token output from the last layer of bert as the input to this module, and then optimally design the adaptive feature modulator AFM to dynamically adjust and optimize the multidimensional representation of input features, enhance the numerical stability, and utilize the learnable activation function and gating mechanism to improve the performance and accuracy of the model. The features of the input data are then efficiently extracted and abstracted by deep convolution and pooling operations and classified and predicted by a fully connected layer.

Adaptive Feature Modulator. In model design, accurate tuning of input features is crucial for performance enhancement. We propose the Adaptive Feature Module (AFM) to improve the adaptability and flexibility of feature representation through a parameterized tuning mechanism. We define three core parameters during model initialization: αfor feature scaling, β for gated intensity regulation, and γ for bias adjustment. The multidimensional design of these parameters enables the model to provide customized adjustments for different levels of features to handle complex feature relationships better. In addition, to enhance the stability of numerical computation, we introduce the learnable epsilon parameter, which can be automatically adjusted according to the changes in training data to optimize the model performance.

The core of AFM lies in its forward propagation logic, which emphasizes the intelligent adaptation of input feature shapes and the flexible choice of feature embedding computation methods. In this process, L1 and L2 paradigms are used to compute feature embeddings. In this paper, the L2 paradigm is chosen. This decision is based on the fact that the L2 paradigm has significant advantages in maintaining directional information and reducing distance bias in high-dimensional data. Moreover, feature embeddings in the L2 paradigm are computed by taking the square root of the sum of squares of the input features, which not only measures the strength of the features but also preserves the interrelationships among them, which is crucial for capturing complex feature interactions.

$$L1\ paradigms\text{: } cmbedding_{l1} = \sum_{d=1}^{D} |x_d| \times a \tag{1}$$

$$L2\ paradigms\text{: } embedding_{l2} = \left(\sum_{d=1}^{D} x_d^2 + \epsilon\right)^{\frac{1}{2}} \times a \tag{2}$$

Subsequently, the normalization factor norm is computed, which is used to adjust the gating strength and ensure that the features are scaled in a controlled manner. The computation of the gating value gate incorporates the learnable activation function act, which learns the optimal nonlinear transformations via a small linear network, further enhancing the expressive power of the model.

$$Ll\ paradigms\text{:}norm = \frac{\gamma}{\sqrt{mean(|embedding|)+\epsilon}} \tag{3}$$

$$L2\ paradigms\text{:}norm = \frac{\gamma}{\sqrt{mean(embedding^2)+\epsilon}} \tag{4}$$

Ultimately, the dynamic regulation of the features is realized by performing an element-by-element product of the input feature x and the gating value gate. This process not only takes into account the spatial dimensions of the features, but also introduces depth dimension conditioning, allowing the model to capture and represent the complexity of the input data in a more fine-grained manner. The unsqueeze and view operations enable individual parameters to operate in new dimensions to accommodate features of different depths, which provides the model with greater flexibility and adaptability.

$$gate = 1 + tanh(act(cmbedding \times norm) + \beta) \tag{5}$$

$$output = x \cdot gate \tag{6}$$

With the adaptive feature modulator AFM is able to adaptively adjust the importance of input features to enhance the model's ability to capture critical information while suppressing unimportant or redundant features.

Deep Convolutional Units. After the adaptive feature modulator AFM dynamically adjusts the weights on the channel features of the input data, the data is then passed through two convolutional layers, which combine the input three-channel data with the features of the BERT_embedding_dim dimension, to output the feature map of 250 channels, a setup that can provide the model with sufficient capability to capture and express rich feature information. Another 2D convolutional layer for further feature extraction on the feature map, keeping the number of channels as 250 and introducing nonlinearity through the ReLU activation function, followed by spreading the feature map into two dimensions, where the first dimension is the batch size and the second dimension is the number of features, and selecting the first 33 features of the spread feature vector based on the feature importance assessment of the model to improve the accuracy and efficiency of the classification. Finally the selected feature vectors are passed through the fully connected layer to get the final classification result logit.

$$\hat{y} = f_{gate}(x) + ReLU\big(Conv2d(x)\big) + Maxpool2d(x) + Linear(x, W, b) \tag{7}$$

4.2 Small-scale feature extractor-BMHA

In this section, we will detail the model branch BMHA proposed in this paper, which aims to enhance the model's ability to capture features at the morpheme level. The module takes as input the output of the token from the last layer of BERT (except for calls labeled tokens), as shown in Figure 1.

Firstly, BMHA adopts a multi-head attention mechanism, which centers on partitioning the input features into multiple heads, 12 heads are selected in this paper, and each head independently learns a different representation subspace of the input. The model first generates queries (Q), keys (K), and values (V) through three linearizations, respectively, to obtain the corresponding representations. Subsequently, Q, K, and V are reshaped and partitioned into multiple heads, each dealing with a portion of the information of the input sequence, using reshaping operations and transposition. The purpose of the reshaping operation is to adjust the dimensionality of each header to $heads\_dim = \frac{embed\_dim}{num\_heads}$, and This scaling operation helps to stabilize the training process. For each head, the dot product attention score between the query and the keys is computed and then scaled to prevent the gradient from vanishing or exploding.

Subsequently, the attention scores are normalized using the softmax function to obtain the attention distribution for each head, the normalized attention distribution is applied to the corresponding value V, the output of each head is obtained by weighted summation, the outputs of all the heads are then combined and projected through a linear layer to obtain the final multi-head attention output, and finally a dropout layer is applied to reduce the overfitting Risk.

$$attn\_probs = softmax(attn\_scores) \tag{8}$$

$$attn\_output = attn\_probs \cdot V \tag{9}$$

$$output = W_o attn\_output + b_o \tag{10}$$

Then, to further process the sequence and enhance the feature sensitivity to the dimensions of the morphemes, the output features of the multi-head attention module are transformed to the appropriate dimensions through a linear layer and then fed into the GRU layer, which consists of a plurality of gating units, each of which comprises an update gate $\mathrm{Z_t}$ and a reset gate r_t, as well as the computation of the candidate hidden states $\hat{h}_t$ and the final hidden states.

$$Z_t = \sigma(W_z \cdot [h_{t-1}, X']) \tag{11}$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, X']) \tag{12}$$

$$\hat{h}_t = tanh(W_h(r_t \cdot x') + U_h h_{t-1}) \tag{13}$$

$$h_t = (1 - Z_t) \cdot h_{t-1} + Z_t \cdot \hat{h}_t \tag{14}$$

Where σ is the sigmoid activation function, the model selects the hidden state of the last time step from the output of the sequence as the final representation of the sequence, and finally, the fully connected layer maps the output of the GRU to the number of target categories to realize the classification task.This design not only enhances the sensitivity of the model to local features but also improves the computational efficiency of the model through parallel processing, which effectively improves the feature extraction of the model at the morpheme level.

4.3 Dynamic Adaptive Fusion Module

In deep learning, model fusion improves prediction performance by combining multiple models. Most traditional fusion methods use a fixed weight averaging or voting mechanism, which ignores the possibility that the weights themselves can also be used as model parameters to be optimized through the training process.

Our proposed adaptive module fusion strategy (Fig. 1-c) introduces two key learnable weight parameters in the model's architecture: self.w_VCDM and self.w_BMHA to optimize the fusion strategy, with initial values of 0.6 and 0.4, respectively. These weights correspond to the outputs of the VCDM and BMHA modules and are automatically adjusted by gradient descent during the training process to achieve the optimal fusion ratio. In the fusion process, the outputs of VCDM and BMHA are first computed independently and then weighted and fused according to the following formula, where VCDM_pre and BMHA_pre represent the classification outputs of the two modules, respectively.

$$pred = self.w\_VCDM * VCDM\_pre \; + \; self.w\_BMHA * BMHA\_pre \tag{15}$$

To ensure the rationality of model fusion, we impose constraints on the weighting parameters to ensure that they sum to 1. This simplifies the weighting parameters' interpretation while ensuring the fusion results' consistency. In the experiments, we verify the effectiveness of the proposed method. The experimental results show that by introducing learnable weights, the model is able to automatically discover the optimal fusion strategy, thus achieving better performance than the traditional fixed-weight fusion method on multiple benchmark datasets.

5 Experiments

5.1 Experimental Setup

Datasets. In order to systematically evaluate the performance of the proposed LBERT-MSDF model, a series of experiments on three authoritative datasets, Data1, Data2 and Data3, are carried out in this study with the aim of performing in-depth comparative analysis and ablation studies. The details of the datasets are summarized in Table 1:

Data1 is based on a selected subset of the THUCNews news text categorization dataset provided by Tsinghua University (THUNLP), containing a total of 200,000 news text records.

Data2 is derived from TNews, the news section of today's headlines, which is a repository containing more than 500,000 texts. Around 300,000 texts were randomly selected from it. Among them, there are 15 different news categories, and 1,000 texts are selected for each category, which are used to construct the validation set and test set respectively.

Data3 is a binary categorization dataset from a software review corpus containing around 12,000 user reviews.DATA3 uses a 6:2:2 ratio to randomly assign samples to the training, validation, and test sets.

Table 1. Statistics on data set details.

Data	classification	aggregate	train	Val	test
Data1	10	20.0000	18.0000	10000	10000
Data2	15	30.0000	27.0000	15000	15000
Data3	2	12000	7200	2400	2400

Evaluation Metrics. In this paper, standard metrics such as accuracy, recall, precision, and F1 score are used for evaluation. The following is a brief explanation of these metrics.

Precision. measures the proportion of samples predicted by the model to be in the positive category that are actually in the positive category:

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{16}$$

Recall. measures the proportion of all samples that are actually positive classes that the model correctly predicts as positive classes. The formula is as follows:

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{17}$$

Accuracy. measures the proportion of all samples that are correctly predicted by the model. The formula is as follows:

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \tag{18}$$

F1 score. is a reconciled average of precision and recall, which balances the two.

$$\mathrm{F1} = 2 \times \frac{\mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} \tag{19}$$

Enviroment Details. The experiments in this paper were conducted on the Ubuntu 22.04 LTS operating system and run with Python 3.8, Pytorch 1.9.1, and CUDA 12.2. The graphics card is 4090 with 24G video memory. The CPU frequency is 2.10 GHz. We summarize the key results of the experiments and the optimal hyperparameters in Table 2.

Table 2. Experimental parameter settings.

Hyper-parameters	DATA1	DATA2	DATA3
Learning rate		1e-5	
Optimizer		AdamW	
Batch size		64	
Dropout	0.5		0.4
Epochs		16	
Max_len	38	40	138
Class_num	10	15	2
Num_heads	12		8
self.w_VCDM		0.6	
self.w_BMHA		0.4	

5.2 Comparative experiments

In this section, the proposed model LBERT-MSDF is trained on the three datasets presented above, and the values of precision, recall, accuracy, and F1 score are obtained. The selection of the comparison models has been thoroughly considered, roughly choosing several models in each of the three aspects: layer considerations, hybrid network models, pre-trained models based on other BERT variants, and the impact of feature fusion strategies. As shown in Tables 3-5, we focus on the comparative analysis of the F1 scores of each model, since the F1 scores are the reconciled averages of precision and recall, which were computed separately for each category and then averaged. The indicators of the target model LBERT-MSDF are generally higher than those of the comparison models, which proves the effectiveness of the models.

Number of Layers Consideration. Since this thesis has done various tests on the output of BERT, two models, BERT-CNN and KBERT-CNN, are selected respectively to compare with the model proposed in this paper to verify the effect of selecting different output layers on the model effect. The KBERT model adopts the output vectors of the last four layers of the BERT model as the inputs of the CNN, and from the experimental results, it can be seen that the KBERT-CNN shows better performance than BERT-CNN on the three datasets, but slightly lower than the model in this paper. The model in this paper is more efficient in terms of training time cost. This improvement is because the BERT model consists of multiple layers of Transformer encoders stacked on top of each other, each of which processes the input text to extract features at different levels. The output of the latter layers contains the richest semantic information, and these layers can capture the global meaning of an entire sentence or paragraph and understand the main idea and deeper meaning of the text. As the layers are stacked, the cost of training time increases. The selection of the number of BERT layers should also focus on the specific model architecture, the network structure proposed in this paper, the last hidden layer is selected, and the specific process can be seen in the experimental part of the ablation experiment section.

Table 3. Results of model comparison in Data1.

Model	Index	Pre	Rec	Acc	F1
KBERT-CNN[15]	Macro avg	0.9367	0.9359	0.9366	0.9363
	Weighted avg	0.9367	0.9359		
BERT-CNN[2]	Macro avg	0.9261	0.9262	0.9264	0.9262
	Weighted avg	0.9261	0.9262		
BERT-BiGRU-TextCNN[16]	Macro avg	0.9325	0.9321	0.9325	0.9323
	Weighted avg	0.9325	0.9321		
BERT-RCNN[17]	Macro avg	0.9235	0.9085	0.9150	0.9159
	Weighted avg	0.9235	0.9085		
TextCNN-RoBERTa[19]	Macro avg	0.9368	0.9370	0.9361	0.9369
	Weighted avg	0.9368	0.9370		
BERT-DPCNN[18]	Macro avg	0.9105	0.9029	0.9105	0.9067
	Weighted avg	0.9105	0.9029		
BERT-MSDF(our)	Macro avg	0.9503	0.9498	0.9503	**0.9503**
	Weighted avg	0.9503	0.9498		

Hybrid Network Models. To better demonstrate the effectiveness of the network structure of this model,three hybrid network models are selected for comparison testing (BERT-BiGRU-TextCNN, BERT-DPCNN, and BERT-RCNN), all of which are based on the BERT pre-training model combined with either a CNN or an RNN structure for feature extraction, aiming to fuse the They are all based on BERT pre-trained models and combined with CNN or RNN structures for feature extraction, aiming to integrate global semantics and local features, but each of them has its focus: BiGRU-TextCNN is long for long-distance dependency but may be redundant, DPCNN is suitable for long text but may ignore word order, and RCNN combines RNN and CNN but has a weak feature interaction. In contrast, the model proposed in this paper takes into account the optimization of sentence-level semantics and word-level feature extraction, which is more advantageous in feature fusion, computational efficiency, and robustness, and performs better in both coarse- and fine-grained tasks by designing input selection in a lightweight manner while avoiding the limitations of a single structure. The specific data are shown in Tables 3-5.

Pre-trained model based on other BERT variants and feature fusion strategy selection. Finally, TextCNN-RoBERTa and KBERT-CNN are tested in comparison with the model proposed in this paper to further prove the progress of the model proposed in this paper.TextCNN-RoBERTa is a kind of hybrid neural network model, RoBERTa is responsible for capturing global semantics and TextCNN is responsible for extracting local features, but it is only simple splicing and slightly insufficient in the feature fusion strategy, and it solves these problems and further improves the effect. Is only simple splicing, and it is slightly insufficient in semantic and long-range dependencies, the model proposed in this paper solves these problems, further improves the results, and the training time is greatly reduced.

Table 4. Results of model comparison in Data2.

Model	Index	Pre	Rec	Acc	F1
KBERT-CNN	Macro avg	0.9268	0.9270	0.9271	0.9269
	Weighted avg	0.9268	0.9270		
BERT-CNN	Macro avg	0.9001	0.9007	0.9007	0.9003
	Weighted avg	0.9001	0.9007		
BERT-BiGRU-TextCNN	Macro avg	0.9171	0.9173	0.9173	0.9172
	Weighted avg	0.9171	0.9173		
BERT-RCNN	Macro avg	0.9001	0.9007	0.9033	0.9032
	Weighted avg	0.9001	0.9007		
BERT-DPCNN	Macro avg	0.9225	0.9221	0.9223	0.9223
	Weighted avg	0.9225	0.9221		
TextCNN-RoBERTa	Macro avg	0.8842	0.8884	0.8872	0.8863
	Weighted avg	0.8842	0.8884		
BERT-MSDF(our)	Macro avg	0.9367	0.9359	0.9360	**0.9363**
	Weighted avg	0.9367	0.9359		

5.3 Ablation experiment

Optimal layer selection. For optimal experimental results, this thesis does not directly choose any combination of BERT outputs as inputs to the model, as different layers of BERT outputs can provide different aspects of the downstream task, with the lower layers (layers 1-4) being closer to the initial text, and more inclined to capture localized, syntactic features. Middle layers (layers 5-8) As the modeling hierarchy deepens, these layers begin to capture a wider range of contextual information and are able to understand more complex linguistic structures and more abstract semantic information. The output of the higher layers (layers 9-12) is usually considered to contain the richest and most abstract semantic information, and these sentences generally capture global information about the entire sentence or paragraph. Therefore, in this paper, we choose the output combinations that currently show better results and higher usage (all hidden layers, four hidden layers after stacking, four hidden layers after connecting, the penultimate layer, and the first layer everywhere, respectively) processed and fed into the

Table 5. Results of model comparison in Data3.

Model	Index	Pre	Rec	Acc	F1
KBERT-CNN	Macro avg	0.8801	0.8803	0.8800	0.8802
	Weighted avg	0.8801	0.8803		
BERT-CNN	Macro avg	0.8896	0.8854	0.7980	0.8875
	Weighted avg	0.8858	0.8031		
BERT-BiGRU-TextCNN	Macro avg	0.8406	0.8628	0.8520	0.8516
	Weighted avg	0.8406	0.8628		
BERT-RCNN	Macro avg	0.8876	0.8876	0.8876	0.8876
	Weighted avg	0.8876	0.8876		
TextCNN-RoBERTa	Macro avg	0.8825	0.8885	0.8802	0.8805
	Weighted avg	0.8825	0.8885		
BERT-DPCNN	Macro avg	0.8807	0.8781	0.8791	0.8794
	Weighted avg	0.8807	0.8781		
BERT-MSDF(our)	Macro avg	0.8962	0.8933	0.8947	**0.8947**
	Weighted avg	0.8962	0.8933		

model, and conduct experiments on the Data1 dataset respectively, to select the inputs that are most suitable for this model.

Table 6. BERT output hidden layer with different selection test results.

Layer	Index	Pre	Rec	Acc	F1
Last Hidden Layer	Macro avg	0.9503	0.9498	0.9503	**0.9503**
	Weighted avg	0.9503	0.9498		
Second to Last Hidden Layer	Macro avg	0.9384	0.9375	0.9375	0.9378
	Weighted avg	0.9384	0.9375		
Sum All 12 Layer	Macro avg	0.9323	0.9318	0.9321	0.9313
	Weighted avg	0.9323	0.9318		
Sum Last Four Hidden	Macro avg	0.9477	0.9478	0.9470	0.9472
	Weighted avg	0.9477	0.9478		
Concat Last Four Hidden	Macro avg	0.9424	0.9422	0.9422	0.9423
	Weighted avg	0.9424	0.9422		

As can be seen from Table 6, the results obtained by using the penultimate hidden layer or taking all the layer outputs stacked and processed are more or less the same, 93.78\% and 93.18\% respectively. However, the training time is greatly increased by taking the output of all layers as input; the results are improved compared to the first two groups by taking the last four hidden layers, stacking, and connecting them as input, which is 94.72\% and 94.23\% respectively. The previous four hidden layers are stacked and connected as input, and the results are improved by 94.72\% and 94.23\%, respectively. Finally, theprevioust hidden layer is selected for processing, as the input of the model, the effect is improved to 95.03\%, and the training time is greatly reduced, reducing the complexity of the calculation, so that the model is more efficient in training, after many experiments, we finally selected the last hidden layer, after processing, as the input of the model.

Ablation experiment.

BERT-DCU. The [cls] token output from Bert's last layer is selected and fed directly into the deep convolutional unit for processing.

BERT-VCDM (AFM+DCU). Addition of AFM module compared to Bert-DCU.

BERT-BMHA. The last layer of Bert outputs vectors other than [cls] tokens are selected as inputs to the BMHA module for processing.

LBERT-MSDF [1]. The outputs of the VCDM model and the BMHA model are summed.

LBERT-MSDF. The model proposed in this paper adopts a dynamic adaptive weighting strategy for dynamically fusing the outputs of the VCDM model and the BMHA model.

Table 7. Table captions should be placed above the tables.

Model	Index	Pre	Rec	Acc	F1
Bert-DCU	Macro avg	0.9368	0.9370	0.9361	0.9369
	Weighted avg	0.9368	0.9370		
Bert-VCDM （AFM+DCU）	Macro avg	0.9407	0.9397	0.9410	0.9402
	Weighted avg	0.9407	0.9397		
Bert-BMHA	Macro avg	0.9361	0.9361	0.9361	0.9361
	Weighted avg	0.9361	0.9361		
Bert-MSDF【1】	Macro avg	0.9467	0.9500	0.9468	0.9468
	Weighted avg	0.9467	0.9500		
Bert-MSDF	Macro avg	0.9503	0.9498	0.9503	**0.9503**
	Weighted avg	0.9503	0.9498		

In this section, we compare the effects of BERT-DCU, BERT-VCDM, BERT-BMHA, and different fusion methods of the outputs of the two modules on the classification results, respectively. According to the experimental results, it can be seen that the addition of the AFM module to the VCDM improves the effect by 2.13\%, which significantly improves the classification performance. This can indicate that the adaptive feature modulator AFM added by the module optimizes the performance of the VCDM module and improves the performance and accuracy of the model. Secondly, the performance of BERT-VCDM and BERT-BMHA are 94.02\% and 93.61\%, and the increased focus on the information of morphemic features through the parallel feature extraction of BERT-BMHA improves the performance of the whole model to 94.68\%, which also proves that by focusing on feature analysis and extraction from the two perspectives of semantic and morphemic, respectively, the performance of the two modules can be effectively features are complemented and enhanced. Subsequent tests on the fusion approach show that the use of a fusion strategy with dynamic adaptive weighting improves the accuracy of the model's classification results by 0.64\% compared to simple summation, which also emphasizes that the optimization of the fusion strategy through the introduction of two key learnable weighting parameters, which are updated by a gradient descent algorithm during the training process as compared to the traditional fixed weights, enables the model to automatically adjust the fusion ratio of the different model outputs, making the model more accurate and efficient for the classification task results.

6 Conclusion

In this study, LBERT-MSDF, a novel dynamic fusion network of multi-scale features, is proposed to address the shortcomings of traditional text categorization models in capturing information at different levels.LBERT-MSDF combines VCDM and BMHA to achieve a macro-semantic understanding of text with micro-lexical structure analysis. By optimizing the inputs and introducing a dynamic weighting mechanism, the model achieves high accuracies of 95.03\%, 93.63\%, and 89.47\% on the three datasets, which outperforms existing techniques.

In the future, we plan to explore pre-trained models, model compression, and acceleration techniques to improve efficiency and maintain performance. LBERT-MSDF will push the field of natural language processing forward.

Acknowledgments. This project consists of:Key R&D Program of Shandong Province, China (2023CXGC010801，2024CXGC010501) .

project ZR2022QG016 supported by Shandong Provincial Natural Science Foundation，the Taishan Industrial Experts Program(NO.tscy20221110).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. J. Zhang, Y. Li, J. Tian, and T. Li, “Lstm-cnn hybrid model for text clas-sification,” (IAEAC) (2018).
2. G. Tanbhir, M. F. Shahriyar, K. Shahed, A. M. R. Chy, and M. A.Adnan, “Hybrid machine learning model for detecting bangla smishing text using bert and character-level cnn,”. [Online]. Available:https://arxiv.org/abs/2502.01518 (2025)
3. Kim, “Convolutional neural networks for sentence classification,”.[Online]. Available: https://arxiv.org/abs/1408.5882 (2014)
4. Z. C. Lipton, J. Berkowitz, and C. Elkan, “A critical review of recurrent neural networks for sequence learning,”. [Online]. Available: https://arxiv.org/abs/1506.00019 (2015)
5. .I. Alshubaily, “Textcnn with attention for text classification,” Available: https://arxiv.org/abs/2108.01921 (2021)

6. .M. Zhang, J. Pang, J. Cai, Y. Huo, C. Yang, and H. Xiong, “Dpcnn-based models for text classification,” Inter-national Conference on Edge Computing and Scalable Cloud (EdgeCom), pp. 363–368. (2023)
7. 7.P. Liu, X. Qiu, and X. Huang, “Recurrent neural network for text classification with multi-task learning,”. https://arxiv.org/abs/1605.05101 (2016)
8. C. B. Vennerød, A. Kjærran, and E. S. Bugge, “Long short-term memory rnn,” Available: https://arxiv.org/abs/2105.06756 (2021)
9. Z. Huang, W. Xu, and K. Yu, “Bidirectional lstm-crf models for sequence tagging,” Available: https://arxiv.org/abs/1508.01991(2015)
10. W. Zhonghao, L. Chenxi, H. Meng, and L. Shuai, “Research on intelli-gent classification method of seismic information text based on bert-bilstm optimization algorithm,” (CCAI), pp.55–59. (2022)
11. word2veclstm,”[Online]. Available: https://arxiv.org/abs/2503.10650
12. Hao Xingyue. BiGRU-CNN text sentiment analysis based on multi-head attention mechanism[J]. Computer Science and Applications, 12(1): 123-134. (2022)
13. Deng Jun, Sun Shaodan, Wang Ruan, Song Xianzhi, Li He. Sentiment evolution analysis of microblog public opinion based on Wor2Vec and SVM[J]. Intelligence Theory and Practice, 43(8):112-19. (2020)
14. K. Yao “Multi-view graph convolutional networks with attention mechanism,” https://www.sciencedirect.com/science/article/pii/S0004370222000480 (2022)
15. Pan, Ning and Yao, Friends Recommendation Based on KBERT-CNN Text Classification Model. (2021)
16. Xinyu Zhang,Zhijie Cai. Text sentiment analysis based on Bert-BiGRU-CNN, (2023)
17. Early Detection of Micro Blog Rumors Based on BERT-RCNN Model. Information studies: Theory & Application. 44(7): 173 (2022)
18. Sun Dandan,Zheng Ruikun et al. Application of BERT-DPCNN model in sentiment analysis of online public opinion[J].Network Security Technology and Application (08):24-27. (2022)
19. S. Salim and O. Lahcen, “A bert-enhanced exploration of web and mobile request safety through advanced nlp models and hybrid architectures,” IEEE Access, vol. 12, pp. 76 180–76 193, (2024)