Computer Science and Information Systems 22(1):133–152 https://doi.org/10.2298/CSIS240630004Z

Multilingual Pretrained based Multi-feature Fusion Model for English Text Classification

Ruijuan Zhang

School of Foreign Languages, Zhengzhou University of Science and Technology
Zhengzhou, 450064, China
ruijzhang2024@163.com

Abstract. Deep learning methods have been widely applied to English text classification tasks in recent years, achieving strong performance. However, current methods face two significant challenges: (1) they struggle to effectively capture long-range contextual structure information within text sequences, and (2) they do not adequately integrate linguistic knowledge into representations for enhancing the performance of classifiers. To this end, a novel multilingual pre-training based multi-feature fusion method is proposed for English text classification (MFFMP-ETC). Specifically, MFFMP-ETC consists of the multilingual feature extraction, the multi-level structure learning, and the multi-view representation fusion. MFFMP-ETC utilizes the Multilingual BERT as deep semantic extractor to introduce language information into representation learning, which significantly endows text representations with robustness. Then, MFFMP-ETC integrates Bi-LSTM and TextCNN into multilingual pre-training architecture to capture global and local structure information of English texts, via modelling bidirectional contextual semantic dependencies and multi-granularity local semantic dependencies. Meanwhile, MFFMP-ETC devises the multi-view representation fusion within the invariant semantic learning of representations to aggregate consistent and complementary information among views. MFFMP-ETC synergistically integrates Multilingual BERT's deep semantic features, Bi-LSTM's bidirectional context processing, and TextCNN local feature extraction, offering a more comprehensive and effective solution for capturing long-distance dependencies and nuanced contextual information in text classification. Finally, results on three datasets show MFFMP-ETC conducts a new baseline in terms of accuracy, sensitivity, and precision, verifying progressiveness and effectiveness of MFFMP-ETC in the text classification.

Keywords: Multi-feature fusion, multilingual pretrained model, English text classification, multi-level structure learning.

1. Introduction

As the scale of the internet expands, vast amounts of data inundate various platforms [21,12,7]. Behind this seemingly chaotic data lies immeasurable value [14]. For example, shopping platforms categorize products into different types, making it easier for users to make purchases. News media accurately classify texts, enabling users to quickly find the information they need, saving time and improving work efficiency. Therefore, the ability to quickly and accurately obtain target information and uncover the potential value behind data has become crucial [6,5]. Text classification technology, as a key solution for

information categorization, has attracted significant attention. Text classification involves mining the semantics of texts and grasping the main topics to categorize texts under predefined labels. Specifically, text classification uses labeled data to train and teach the model. Through training and learning, the model learns certain classification rules, and finally, an established classifier is used to predict and categorize unknown texts. As a fundamental technology in natural language processing, text classification is the cornerstone of many NLP tasks. It is widely applied in fields such as news classification, spam filtering, sentiment analysis, and public opinion analysis [22].

In recent years, deep learning-based text classification methods have emerged one after another, significantly improving classification efficiency [13,2,30,9]. Neural network models such as convolutional neural networks, self-attention models, generative adversarial networks, and recurrent neural networks have been widely applied to text classification tasks. Compared to traditional statistical learning methods like support vector machines (SVMs), deep learning models generally exhibit superior performance in classification. This is because deep learning models can automatically extract high-level features from large datasets, reducing the reliance on manual feature engineering while possessing strong representation learning capabilities. For instance, Multilingual BERT, based on the self-attention mechanism and large-scale pretraining, can capture contextual information, handle long-distance dependencies, and enhance the generalization ability of the model through multitask learning.

Despite the strong performance of deep learning-based text classification models in various scenarios, challenges remain in capturing semantic information related to long-distance dependencies within texts. Researchers have proposed several innovative methods to address this issue. For instance, Mundra et al. developed the hierarchical attention network, which improves text structure understanding by introducing two levels of attention mechanisms—sentence and word levels—using bidirectional recurrent neural networks for encoding [18]. Liu et al. introduced the MEANI model, which employs an attention mechanism to integrate emotional language features into the neural network, thereby enhancing the model ability to handle complex emotional expressions [16]. While self-attention-based models have made significant progress in addressing long-distance dependency issues, they have not entirely solved the problem. A key limitation is that many models focus primarily on sentence or document-level processing without explicitly incorporating linguistic knowledge, which is essential for grasping nuanced semantic relationships across different contexts. This shortcoming can lead to suboptimal performance in certain scenarios. Take, for example, the sentence: “Although the plot of this film may seem somewhat monotonous and indistinct, it uniquely captivates our perception in a manner that few contemporary films achieve.” At first glance, the sentence appears to criticize the film’s plot. However, considering the context—especially the phrase “in a manner that few contemporary films achieve”—the sentence is actually offering high praise. If a model fails to incorporate linguistic knowledge effectively, it might misclassify this sentence as a negative review, illustrating the limitations of current self-attention models in capturing subtle semantic relationships.

To this end, a novel multilingual pre-training based multi-feature fusion method is proposed for English text classification (MFFMP-ETC). Specifically, MFFMP-ETC consists of the multilingual feature extraction, the multi-level structure learning, and the multi-view representation fusion. Specifically, multilingual feature extraction utilizes the power

of Multilingual BERT to extract rich semantic features from English texts. By capturing both contextual and nuanced language details, it provides a solid foundation for precise text classification, which ensures that even subtle linguistic cues are effectively identified and leveraged. Then, multi-level structure learning combines the strengths of Bi-LSTM and TextCNN to capture both global and local features of the text. The Bi-LSTM focuses on understanding long-term dependencies in both directions, ensuring that contextual relationships within the text are thoroughly explored. Meanwhile, TextCNN is responsible for identifying important local features using convolutional techniques, capturing finer details that might be missed by global models. Furthermore, multi-view representation fusion merges the features extracted by the previous components into a unified and comprehensive representation. By fusing global and local insights, it creates a more robust and holistic understanding of the text which enhances the model sensitivity to both the overall context and detailed features, resulting in superior classification performance. Finally, experiment results prove the model advantages in recognizing complex semantic structures and enhancing classification precision, setting a new baseline for English text classification tasks.

The contributions of MFFMP-ETC are threefold:

- A multi-level structure learning within the deep multilingual feature extraction architecture is proposed via modelling bidirectional contextual semantic dependencies and multi-granularity local semantic dependencies, which captures global and local structure information of English texts.
- An invariant semantic learning is devised to aggregate consistent and complementary information among representations of views for obtaining a more robust and holistic understanding of the text which enhances the model sensitivity to both the overall context and detailed features, resulting in superior classification performance.
- Experiment results illustrate the efficacy of MFFMP-ETC in comparison to existing methods, highlighting its superior accuracy across three English text classification datasets.

Next, Section 2 reviews related research on text classification methods based on deep learning, including methods based on convolutional neural networks, methods based on recurrent neural networks, methods based on feedforward neural networks, methods based on graph neural networks, and methods based on pre-trained language models. Section 3 provides a detailed introduction to the structure and working principles of the MFFMP-ETC model, including the input layer, multilingual BERT pre-trained language model layer, (Bi-LSTM layer, TextCNN layer, multi-feature fusion layer, and classifier layer. Section 4 presents the experimental results on the MR, SST-2, and CoLA datasets, demonstrating the significant improvement in classification accuracy of the MFFMP-ETC model compared to existing models. Finally, Section 5 summarizes the main contributions of this paper and discusses potential future research directions, including exploring more complex attention mechanisms, integrating domain knowledge, and applying the model to other related tasks to further enhance text classification performance.

2. Related Works

Deep learning-based text classification model refer to a model that utilizes deep neural network architectures to extract features from textual data and classify the text. This model

learns from a large amount of training data to automatically capture semantic information in the text and categorize it into one or more predefined classes. It is primarily divided into five branches: methods based on convolutional neural networks (CNNs), methods based on recurrent neural networks (RNNs), methods based on feedforward neural networks (FNNs), methods based on graph neural networks (GNNs), and methods based on pre-trained language models (PLMs).

Methods based on CNNs: CNNs are initially used for image processing. With further research, they began to be applied in the field of text classification. Kim et al. proposed the classic TextCNN model, which was the first to combine CNNs with text classification tasks [2]. The TextCNN first utilized word2vec for word vector initialization, then used multiple kernels of different sizes to extract key information from sentences, helping the model better capture local features. Finally, the model fed the acquired data into a fully connected layer for final classification and output. Although this model can better capture text features, it loses lexical order and positional information during convolution and pooling operations, and can only capture local word order information, which is detrimental to the final classification results. In addition to the TextCNN model, there are also classic text classification models such as CharCNN [30], DPCNN [9], and CCRCNN [27]. CharCNN transforms the input text into individual characters, representing the text using strings without relying on syntactic and semantic features of the text. This approach has a good fault tolerance rate and can improve classification accuracy. The DPCNN model further refines and improves upon TextCNN by increasing the network depth to enhance text feature extraction, thereby improving text classification accuracy. The CCRCNN model is suitable for short text classification. It extracts contextual features of the text through the network, uses attention mechanisms to capture contextual concepts, and integrates these conceptual features, thereby strengthening the model's ability to capture semantic information and improving classification accuracy.

Methods based on RNNs: Text is sequential in nature, and understanding it requires connecting words in a specific sequence rather than interpreting each word in isolation. For example, the meaning of "I," "eat," and "banana" changes depending on their order. RNNs are designed to handle sequential data, making RNN-based text classification models widely applicable [19]. RNNs can capture positional information and long-distance dependencies in sequences, enhancing the model's ability to capture semantic information in text. However, RNNs suffer from the issues of gradient vanishing and exploding, which prevent parallel computation and result in high computational costs. Despite these challenges, many excellent RNN-based text classification models have been developed. TextRNN is one of the classic RNN-based text classification models. It is a multi-task learning model that can be used in scenarios with limited sample data and has achieved good results on many datasets. Other classic RNN-based text classification models include the MT-LSTM model [17] and the HAN model [24]. The MT-LSTM model leverages LSTM, a variant of RNN, by categorizing the hidden states of LSTM into several groups and activating or updating them at different intervals, making the model suitable for long text classification. The HAN model divides the text into sentences, encodes words and sentences using bidirectional LSTMs, and then employs an attention mechanism to strengthen feature capture. Finally, it uses a softmax layer for text classification prediction, achieving excellent results in long text classification tasks. The semi-supervised text classification model based on bidirectional LSTM, proposed in 2019, integrates various

loss functions such as cross-entropy, which also enhances the accuracy of text classification to some extent [20].

Methods based on FNNs: Models based on feedforward neural networks are also applied to text classification tasks. By modifying word embeddings, feedforward neural network models enhance the extraction of text features, significantly improving text classification accuracy. One of the most classic models is the fastText model [10]. This model introduces the concept of n-grams, summing and averaging the input word vectors before feeding them into a softmax layer for classification. By converting multi-class tasks into binary classification tasks, the model maintains low complexity and few parameters, enabling fast and efficient text classification. Additionally, the SWEM model also demonstrates excellent performance in long text classification [23]. This model captures the maximum value of each dimension in word embeddings through max pooling, thereby extracting key feature information from each word embedding. It then performs hierarchical pooling by averaging all windows, and finally uses global max pooling to sample text features, capturing the most prominent text features. This process enhances the model's ability to extract text features, allowing it to achieve good results in text classification.

Methods based on GNNs: To address the limitations of traditional deep learning models in long-distance information transfer and comprehensive text semantics extraction, recent research has shifted attention to Graph Neural Networks (GNNs). GNNs were initially designed for applications in graph-structured data. Nowadays, with deeper research, GNNs have been applied to the field of text classification. Graph Neural Networks can define relationships between multiple concepts and preserve global structural information. They can transform text classification tasks into graph node classification tasks. Due to their unique properties, many classic text classification models based on GNNs have been developed. Firstly, the graph-CNN model was proposed to convert text into graphs before classification [3]. The model transforms text into a text graph and uses graph convolution operations to capture long-distance text semantics, enhancing the model's ability to capture semantic information. TextGCN is another classic text classification model based on GNNs [28]. This model constructs a text graph for the corpus based on word co-occurrence and semantic relationships between words. The special properties of the graph structure allow long-distance information transmission, improving the accuracy of text classification. Zhang et al. proposed a heterogeneous graph neural network based on transformers [29]. This model introduces additional structural encoding to account for node heterogeneity, and the integration within the transformer allows for learning node representations. Lin et al. proposed BertGCN, which integrates BERT with GCN to capture text features [15]. However, this model has a narrow focus on text feature information and does not consider text features from multiple perspectives. Xie et al. proposed the TV-GAE model, which captures feature information using GNNs [26]. This model integrates a topic model into the graph structure to capture semantic information between text and words, enhancing the model's ability to learn text semantics and improving text classification accuracy. Wang et al. proposed GLHG, a new graph construction method capable of distinguishing different word documents, which is a new cross-language heterogeneous graph neural network model [25]. Li et al. proposed the TextGTL model, which uses a non-heterogeneous graph construction method [11]. This model constructs semantic text graphs, context text graphs, and syntactic text graphs, and jointly trains multiple graphs to capture significant feature information.

Methods based on PLMs have also achieved great success in text classification tasks by enhancing the extraction of text features through extensive pre-training. One of the most classic models is the BERT pre-trained model, proposed in 2018. BERT is trained on a large amount of unlabeled data to extract general features, which are then used to complete classification tasks [4]. Specifically, the BERT model utilizes the encoder structure of the Transformer and is trained through two tasks. The first task is Masked Language Modeling (Masked LM), where some tokens are masked and the model predicts the masked tokens. The second task is Next Sentence Prediction, which involves randomly selecting sentence pairs (sentence A and sentence B) and predicting whether sentence B follows sentence A. This approach allows BERT to deeply learn word-level and sentence-level features, improving its performance in subsequent classification tasks. In addition to BERT, VAMPIRE is another classic pre-trained language model [8]. This model first inputs unlabeled text into a variational autoencoder (VAE) to learn general features through pre-training. The data is then fine-tuned within the VAE model to obtain corresponding word vector representations, which are then concatenated with GloVe word vectors. Finally, the text classification is completed using an encoder and an MLP. MitText, proposed in 2021, is another pre-trained language model [1]. This model uses BERT to predict mixed labeled sample data, generating pseudo-labels. It then performs TMix training and combines TMix with other data, enhancing the data through back-translation, thereby improving the model's classification accuracy.

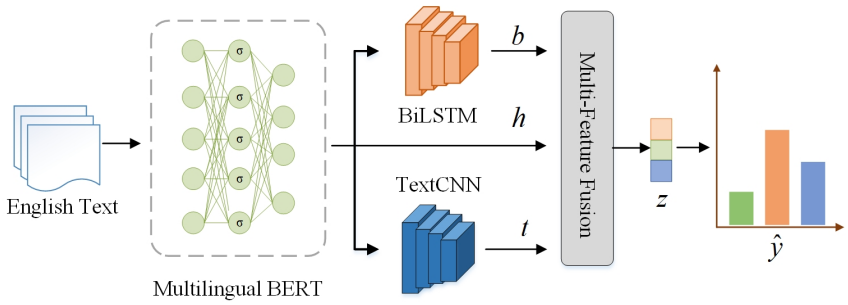


Fig. 1. The illustration of MFFMP-ETC, containing the multilingual feature extraction, the multi-level structure learning, and the multi-view representation fusion

MFFMP-ETC stands out from traditional text classification methods due to its unique integration of advanced techniques. Unlike CNNs that primarily capture local features and often lose positional information, MFFMP-ETC combines the strengths of Multilingual BERT, Bi-LSTM, and TextCNN to address both local and global semantic contexts. While RNNs like TextRNN handle sequential data and long-distance dependencies, they struggle with issues like gradient vanishing, which MFFMP-ETC overcomes through its multi-feature fusion approach. FNNs, such as fastText, efficiently process text but may lack depth in capturing complex semantic relationships. GNNs enhance long-distance information transfer through graph structures but may not fully leverage linguistic knowledge. Pre-trained Language Models PLMs like BERT provide deep contextual features

but often do not integrate other feature extraction techniques. MFFMP-ETC synergistically integrates Multilingual BERT's deep semantic features, Bi-LSTM's bidirectional context processing, and TextCNN local feature extraction, offering a more comprehensive and effective solution for capturing long-distance dependencies and nuanced contextual information in text classification.

3. Multilingual Pretrained based Multi-feature Fusion Model for English Text Classification

A novel English text classification model is proposed via integrating multi-view features within the multilingual pre-training optimization framework (MFFMP-ETC), as shown in Fig. 1. MFFMP-ETC consists of the multilingual feature extraction, the multi-level structure learning, and the multi-view representation fusion. The main mathematical notations in MFFMP-ETC are listed in Table 1.

Table 1. Frequently used notations

Notations	Description
x_i	the i-th word in the sentence.
h_i	the i-th word embedding vector
b	the feature generated by BiLSTM
t	the feature generated by TextCNN
z	the fusion feature
MLP	the multi-layer perceptron
L_{pos}	positive Pair Loss
L_{neg}	negative Pair Los
L_{cross}	the cross-entropy loss
λ,β	the balance coefficients

3.1. Multilingual feature extraction

In general, multilingual pre-trained encoders address the challenge of capturing long-range contextual semantic information within text sequences by leveraging their training across multiple languages. These models have learned to understand diverse sentence structures and linguistic patterns, enabling them to better capture long-distance dependencies in text. By being exposed to languages with varying syntactic and grammatical rules, multilingual encoders develop a more flexible and comprehensive approach to understanding context, which enhances their ability to model relationships between distant words in a sentence.

Hence, MFFMP-ETC utilizes the multilingual BERT to extract features from English texts for enhancing representation discriminability. Firstly, the preprocessing is conduced on each sentence, that is, a [CLS] token, indicating the beginning of the sentence, is added to the start, and a [SEP] token, indicating the end of the sentence, is inserted at the end. After preprocessing, the input English sentence is converted into three types of input vector

embeddings: word embeddings, segment embeddings, and position embeddings. The final input is the sum of these three types of embeddings. In this process, segment embeddings are mainly used to differentiate between pairs of sentences by connecting the sentences in the input text using the [CLS] token, which helps determine the order of two different sentences during pre-training. Position embeddings are primarily used to distinguish the semantic differences of words in different positions within the text sequence.

Specifically, given an input English sentence x composed of k words, which can be formalized as:

$$x = [x_1, x_2, x_3, \ldots, x_k] \tag{1}$$

where x_i represents the i-th word in the sentence.

In the experiments, the masked language model pre-training strategy is used to enable the model to learn contextual features of sentences. In this task, a random 15% of the words in the text are masked with a special token, [MASK], and the model is then tasked with predicting the masked words based on the final hidden output vectors obtained through softmax functions. Here's an example of the masking operation:

Original sentence: After watching the movie, I think it is better than the one I saw last week.

Masked sentence: After watching the movie, I think it is [MASK] than the one I saw last week.

However, since the input vectors do not include the [MASK] token as mentioned earlier, a mask strategy is needed to address this issue. For words to be randomly masked with the [MASK] token, the mask strategy is as follows: (1) 80% of the words are directly replaced with [MASK]. (2) 10% of the original words are replaced with any other word. (3) The remaining 10% are left unchanged. This strategy ensures varied input for the model training, enhancing its understanding of context. For instance: (1)"After watching the movie, I think it is [MASK] than the one I saw last week." (2) "After watching the movie, I think it is no than the one I saw last week." (3) "After watching the movie, I think it is better than the one I saw last week."

After pre-training, using the sum of three embedding vectors generated by Multilingual BERT as the input for both the BiLSTM layer and the TextCNN layer:

$$h = [h_1, h_2, h_3, \ldots, h_k] \tag{2}$$

where the dimension of h_i is 768.

3.2. Multi-level structure learning

To capture multi-level structure information for learning robust and comprehensive fusion representations, MFFMP-ETC integrates BiLSTM and TextCNN into a multi-feature fusion module. TextCNN is responsible for extracting local features, such as key phrases and n-gram patterns, while BiLSTM captures global contextual information and long-range dependencies within the text. By combining these two approaches, the fusion module generates a rich representation that balances local and global structures, leading to improved generalization and robustness across different types of text data.

BiLSTM feature extraction: BiLSTM involves processing the text in both forward and backward directions, capturing global structure features from the entire sequence. The

BiLSTM learns long-range dependencies and contextual relationships by using hidden state vectors at each time step. Specifically, the input to the BiLSTM network at time step t is the concatenation of the hidden state from the previous time step h_{t-1} and the current input vector x_t. This enables the BiLSTM to integrate information from both past and future contexts, refining the structure features produced by Multilingual BERT. As a result, BiLSTM captures global dependencies within the text:

$$\vec{h}_t = \sigma(W_t[\vec{h}_{t-1}, x_t] + b) \tag{3}$$

The output vector of the hidden state of the LSTM network from back to front is:

$$\overleftarrow{h}_t = \sigma(W_t[\overleftarrow{h}_{t-1}, x_t] + b) \tag{4}$$

where W_t and b represent the weight matrix and bias vectors of the forward and backward LSTM networks. $\vec{h}_{t-1}$ and $\overleftarrow{h}_{t-1}$ denote the forward and backward hidden state vectors at time step $t-1$, respectively. Finally, at time step t, the hidden state vector of the Bi-LSTM layer is the concatenation of the forward and backward hidden state vectors:

$$b_t = [\vec{h}_{t-1}, \overleftarrow{h}_{t-1}] \tag{5}$$

After processing all time steps, a collection of hidden layer vectors that encapsulate long-range contextual semantic information is obtained, denoted as:

$$B = [b_1, b_2, b_3, \ldots, b_k] \tag{6}$$

TextCNN feature extraction: TextCNN excels at extracting local features and captures different N-gram features through convolution windows of various sizes. A piece of text contains local semantic features of different granularities, thus necessitating the extraction of features at different scales. By designing convolution kernels of various sizes to extract text information and integrating features of different granularities, a comprehensive local feature representation can be achieved. Using the convolution structure of TextCNN, local features of the text can be extracted, facilitating text classification. The outputs of all encoders in multilingual BERT are fed into the TextCNN layer. By adjusting the sizes of convolution kernels, local text features are obtained at different widths. Convolution operations are performed on the text sequence using convolution kernels with sizes of 2, 3, and 4. The resulting vectors are passed through the ReLU activation function to capture important sentence information. The feature vectors obtained after pooling are concatenated to form the output of the TextCNN layer.

Convolution operation formula: For a text sequence $h = [h_1, h_2, h_3, \ldots, h_k]$ and a convolution kernel of size s, the feature c_i extracted by the i-th convolution kernel is:

$$c_i = f(\mathbf{w} \cdot \mathbf{h}_{i:i+s-1} + b) \tag{7}$$

where $\mathbf{h}_{i:i+s-1}$ denotes the concatenation of word vectors from position i to $i+s-1$, $\mathbf{w}$ is the weight matrix, b is the bias, and f is the activation function (e.g., ReLU). After the convolution operation, max-pooling is applied to extract the most significant feature from each feature map. For a feature map $\mathbf{c} = [c_1, c_2, \ldots, c_{n-k+1}]$, the pooled feature p is:

$$p = \max([c_1, c_2, \ldots, c_{n-k+1}]) \tag{8}$$

In the MFFMP-ETC, Using multiple convolution kernels of different sizes (e.g., 2, 3, and 4), the final output feature vector t of the TextCNN layer is obtained by concatenating the outputs from all convolution and pooling operations:

$$t = concat[p_2, p_3, p_4] \tag{9}$$

where concat[] denotes concatenating function.

3.3. Multi-view representation fusion

The benefits of contrastive learning for learning fusion representations lie in its ability to effectively utilize information from multiple views, thereby enhancing the model performance and generalization ability. By merging representations from different views, richer and more accurate semantic information can be obtained, which helps to improve the quality of text representations and the effectiveness of the model.

The fusion representation is obtained through an MLP fusion layer, where the input consists of the concatenation of three features. This process can be expressed by the following formula:

$$z = f(h, t, b) = \text{MLP}\left([\text{Concatenate}(h, t, b)]\right) \tag{10}$$

Here, h, t, and b represent the feature representations from the multilingual BERT feature extraction, the textCNN feature extraction, and the BiLSTM feature extraction, respectively. $[\cdot]$ denotes vector concatenation, and MLP represents a multi-layer perceptron.

In the contrastive learning framework, the goal is to maximize the similarity between the fused representation z and the feature representations while minimizing the similarity between these representations.

$$\mathcal{L}_{\text{pos}} = -(\cos(z, h) + \cos(z, t) + \cos(z, b)) \tag{11}$$

$$\mathcal{L}_{\text{neg}} = \cos(h, t)) + \cos(h, b) + \cos(t, b) \tag{12}$$

$$\mathcal{L}_{\text{contra}} = \mathcal{L}_{\text{pos}} + \lambda \mathcal{L}_{\text{neg}} \tag{13}$$

where λ is the parameter for balancing $\mathcal{L}_{\text{pos}}$ and $\mathcal{L}_{\text{neg}}$. $\cos(,)$ denotes the similarity function.

Meanwhile, MFFMP-ETC uses a fully connected classification layer as the classification head to transform the fusion representation z into scores for each class, and then uses the softmax function to convert these scores into probabilities.

The output y of the classification head can be represented as:

$$y = \text{softmax}(W_{\text{class}} z + b_{\text{class}}) \tag{14}$$

Here, W_{class} is the weight matrix of the the classification head, b_{class} is the bias vector, and softmax is the softmax function. y is a vector of size K, representing the probability distribution over each class.

Then, the cross-entropy loss for K classes is utilized to achieve pattern mining of English text, which can be expressed as:

$$\mathcal{L}_{cross} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} y_{i,k}\log(\hat{y}_{i,k}) \tag{15}$$

Where N is the number of samples, $y_{i,k}$ is the true label indicating whether the i-th sample belongs to the k-th class, and $\hat{y}_{i,k}$ is the predicted probability by the model for the i-th sample belonging to the k-th class. This process transforms the fused representation z into a probability distribution over each class, enabling text classification.

3.4. Overall Objective Function

The MFFMP-ETC model employs an integrated objective function that combines three distinct types of losses to optimize its performance comprehensively. The overall objective function incorporates the following components:

$$\mathcal{L} = \mathcal{L}_{\text{pos}} + \lambda\mathcal{L}_{\text{neg}} + \beta\mathcal{L}_{\text{cross}}, \tag{16}$$

where λ and β are balancing parameters that control the influence of the negative pair loss and cross-entropy loss, respectively.

The choice of this integrated objective function is driven by several key advantages:

- Enhanced semantic representation: By incorporating the positive pair loss, the model benefits from a more comprehensive and enriched semantic representation. This fusion of information from multiple feature sources allows the MFFMP-ETC model to better capture and represent the nuanced meanings within the text, enhancing its overall expressiveness.
- Reduced information redundancy: The negative pair loss contributes to minimizing redundancy by maximizing the mutual information between the fused representation and individual viewpoints. This reduction in redundancy not only improves the efficiency of the model but also enhances its generalization capability, making it more effective in diverse classification scenarios.
- Improved model robustness: The combination of contrastive learning losses helps the model handle noise and errors more effectively. By focusing on both positive and negative pairs, and aligning predictions with true labels through cross-entropy loss, the model becomes more robust and stable, which is crucial for achieving reliable performance in practical applications.

Overall, this multi-faceted loss function allows MFFMP-ETC to balance feature fusion, differentiation, and classification accuracy, leading to significant performance improvements in English text classification tasks compared to traditional single-loss or less integrative approaches.

4. Experiments

4.1. Set up

Dataset and metric: The following three common English text datasets, i.e., MR dataset, SST-2 dataset, and CoLA dataset, are primarily employed for training and testing MFFMP-ETC:

- MR dataset is constructed based on brief movie review texts. The training set mainly includes 5,331 negative samples and 3,610 positive samples.
- SST-2 dataset is a variant of the MR dataset. It should be noted that very positive and affirmative review texts are labeled as positive samples, while negative and extremely negative review texts are labeled as negative samples. Overall, the training set is divided into 3,310 negative samples and 3,610 positive samples.
- CoLA dataset is a dataset used for binary single-sentence classification tasks, which contains 8,551 training data and 1,043 test data, totaling 6,744 positive samples and 2,850 negative samples. The average text length of the dataset is 7.7 words. Since the test set of CoLA is not annotated, this paper allocates 5% of the samples from the training set as the validation set and uses the original validation set as the test set.

For evaluation metrics, MFFMP-ETC mainly uses accuracy, sensitivity, and precision to measure and assess the effectiveness and performance of MFFMP-ETC and comparative models. Accuracy is calculated using the following formula:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{17}$$

where TP, TN, FP, and FN represent the number of positive samples correctly identified, negative samples correctly identified, negative samples incorrectly identified as positive, and positive samples incorrectly identified as negative, respectively. Sensitivity is calculated using the following formula:

$$\text{Sensitivity} = \frac{TP}{TP + FN} \tag{18}$$

This metric measures the proportion of actual positives that are correctly identified by the test. Precision is calculated using the following formula:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{19}$$

This metric measures the proportion of positive identifications that are actually correct.

Implementation Details: In the experiments, MFFMP-ETC is meticulously constructed on the NVIDIA GeForce RTX 3080 Ti graphics card, leveraging the capabilities of the PyTorch framework within the Visual Studio Code as the development environment, where Python version 3.6.8 is deployed. Word2Vec vectors, each with a dimension of 300, are employed to represent textual data, accommodating a maximum text length of 150 characters.Bi-LSTM network layer within MFFMP-ETC is tailored with 16 nodes to effectively capture the dynamics of sequential information. MFFMP-ETC undergoes a structure training regimen comprising 10 epochs and a batch size of 32, parameters

chosen to balance thorough training with the risk of overfitting. The Adam optimizer is engaged for its adaptive learning rate mechanism, initiated with a learning rate of 0.001 and accompanied by a decay rate of 0.1 to strategically taper the learning rate throughout the training process. To augment the model generalization capabilities and mitigate overfitting, Dropout with a 0.3 probability is seamlessly integrated into the MFFMP-ETC model training phase. This method introduces an element of randomness by deactivating a subset of neurons during training, compelling the model to develop a more diverse and robust set of features. Furthermore, MFFMP-ETC benefits from the incorporation of word embeddings derived from the pre-trained multilingual BERT model, which, with a dimensionality of 768, provides a comprehensive semantic representation. This integration allows MFFMP-ETC to capitalize on the nuanced language understanding expertise acquired by BERT during its pre-training.

Table 2. Comparison results of methods on different datasets in terms of accuracy (Acc), precision (Prec), and sensitivity (Sens)

Method	MR dataset			SST-2 dataset			CoLA dataset		
	Acc	Prec	Sens	Acc	Prec	Sens	Acc	Prec	Sens
SVM	0.745	0.715	0.744	0.794	0.777	0.789	0.572	0.499	0.556
MLP	0.759	0.726	0.749	0.808	0.810	0.799	0.608	0.612	0.576
CNN-non-static	0.815	0.815	0.812	0.872	0.877	0.874	0.617	0.613	0.631
LSTM	0.804	0.799	0.800	0.859	0.866	0.864	0.612	0.632	0.605
Bi-LSTM	0.813	0.812	0.812	0.882	0.888	0.898	0.625	0.666	0.625
Multilingual Bert	0.816	0.815	0.822	0.912	0.910	0.901	0.811	0.804	0.802
MFFMP-ETC	**0.862**	**0.866**	**0.859**	**0.915**	**0.920**	**0.919**	**0.832**	**0.818**	**0.820**

4.2. Comparison with baselines

Comparison methods: To thoroughly evaluate the capability of MFFMP-ETC to accurately capture both local and global advanced contextual semantic information, a series of comparative and evaluative experiments are conducted with MFFMP-ETC against several benchmark models. In addition to the original Bert pre-trained language model being directly applied to English text classification tasks, several relevant deep learning network models are selected for comparative experiments. The detailed information on the selected benchmark models is described as follows: SVM: To ensure the comprehensiveness of the comparative experiment, the traditional classification model SVM is specifically chosen as a benchmark for comparison in MFFMP-ETC. MLP: The multi-layer perceptron, a traditional model, has two hidden layers comprising 512 and 100 hidden units, respectively. This perceptron is utilized with bag-of-words vectors that are weighted by term frequency (TF). CNN-non-static: The input vectors for this model are kept consistent with those of the MLP, with the model original training parameters remaining unaltered. LSTM: The hidden layer output vectors of the Bert model are served as inputs to a unidirectional Long Short-Term Memory network. Bi-LSTM: The bidirectional LSTM network is provided with inputs that are consistent with those of the unidirectional network. Multilingual Bert: In the experiments, the original pre-trained multilingual Bert model is utilized. To

guarantee that the comparative experiments represent a fair comparison, all models have been trained from scratch.

Comparison results: As shown in Table 2, MFFMP-ETC outperforms all other models across all datasets and evaluation metrics, demonstrating advantage in extracting both local and global high-level contextual semantic information. On the MR dataset, MFFMP-ETC achieved the highest scores among all models in terms of Acc, Prec, and Sens, with values of 0.862, 0.866, and 0.859, respectively. On the SST-2 dataset, MFFMP-ETC once again demonstrated the best performance across all evaluation metrics, with Acc, Prec, and Sens, being 0.915, 0.920, and 0.919, respectively. On the CoLA dataset, the MFFMP-ETC performance was also the best among all compared models, with an Acc of 0.832, a Prec of 0.818, and a Sens of 0.820. The reasons are twofold: (1) Utilization of multilingual pretrained models. MFFMP-ETC leverages a multilingual BERT model as one of its core components. This pretrained model has been trained on a variety of languages, capturing cross-linguistic semantic information, thereby enhancing the model's comprehension and classification capabilities for English texts. This cross-linguistic semantic understanding is a capability that traditional monolingual models lack, giving MFFMP-ETC an advantage when dealing with English texts that are multilingual or have complex semantic structures. (2) Multi-feature fusion strategy. MFFMP-ETC employs a multi-feature fusion strategy, effectively combining the deep semantic features from the BERT model, the bidirectional contextual features from Bi-LSTM, and the local n-gram features from TextCNN. This fusion strategy not only enhances the model's ability to capture a wide range of textual information but also improves the accuracy and robustness of classification through the complementary nature of the features.

In addition, in the realm of text classification, the ability to discern and utilize semantic information is paramount. The Multilingual Bert model's success over traditional models such as SVM, MLP, and LSTM is a testament to its advanced capability to capture and process linguistic nuances. This pre-trained model, with its exposure to a diverse range of languages, has honed a deep understanding of language constructs that transcends the limitations of models trained solely on local context. The Bi-LSTM model's enhancement over its unidirectional counterpart is particularly noteworthy. By processing information in both forward and backward sequences, Bi-LSTM is able to develop a more comprehensive representation of the text, thus enhancing its predictive accuracy. This bidirectional capability is crucial for understanding the context in which words are used, as the meaning of a sentence can be significantly altered by the words that precede or follow it. Despite the Bi-LSTM model showing a slight dip in performance when compared to the CNN-non-static model on the MR dataset, this is not indicative of a weakness. Instead, it highlights the potential for hybrid models that can leverage the strengths of various architectures. The CNN-non-static model, with its ability to capture local features through convolution operations, complements the Bi-LSTM's contextual prowess. The combination of the Bi-LSTM with the Multilingual Bert model is a case in point. This synergy not only bolsters the model's ability to express and extract semantic information but also significantly amplifies the overall performance of the classification task. The Multilingual Bert model's pre-training on a vast corpus of text endows it with a rich vocabulary of linguistic patterns and structures, which, when combined with the Bi-LSTM's temporal insights, results in a model that is both sensitive to local features and attuned to broader contextual elements.This integrated approach to text classification is a step to-

wards more sophisticated models that can handle the intricacies of natural language with greater finesse. It opens up avenues for further research and development, encouraging the exploration of additional hybrid models and the refinement of existing architectures. As we continue to push the boundaries of what is possible with text classification models, the fusion of diverse methodologies will undoubtedly play a key role in shaping the future of natural language processing.

4.3. Ablation Study

Table 3. Ablation experiments of each component on MR

L_{pos}	L_{neg}	L_{cross}	Accuracy	Precision	Sensitivity
		√	0.798	0.780	0.794
√		√	0.842	0.842	0.840
	√	√	0.814	0.812	0.820
√	√	√	**0.862**	**0.866**	**0.859**

Three ablation experiments about L_{pos}, L_{neg}, and L_{cross} are conducted to prove the effectiveness of each component. Specifically, (1) MFFMP-ETC utilizes L_{cross} to train the network. This configuration serves as a baseline where only the cross-entropy loss is used. The results show an accuracy of 0.798, precision of 0.780, and sensitivity of 0.794. This indicates the performance when contrastive components are not used. (2) MFFMP-ETC utilizes L_{pos} and L_{cross} to train the network. In this setup, the positive sample pair loss is included along with the cross-entropy loss, resulting in improved performance with an accuracy of 0.842, precision of 0.842, and sensitivity of 0.840. This demonstrates the effectiveness of maximizing the similarity between the fused representation and feature representations. (3) MFFMP-ETC utilizes L_{neg} and L_{cross} to train the network. Here, the negative sample pair loss is included along with the cross-entropy loss, leading to an accuracy of 0.814, precision of 0.812, and sensitivity of 0.820. This shows the benefit of minimizing the similarity between different feature representations. These results indicate that both L_{pos} and L_{neg} contribute significantly to the model's performance. Including all three components (L_{pos}, L_{neg}, and L_{cross}) achieves the best results, demonstrating the effectiveness of the proposed framework.

As shown in Table 3, there are three conclusions: (1) The inclusion of L_{pos} significantly improves performance. Comparing the results with and without L_{pos}, we observe that accuracy increases from 0.798 to 0.842, precision from 0.780 to 0.842, and sensitivity from 0.794 to 0.840 when L_{pos} is added. This demonstrates the effectiveness of maximizing the similarity between the fused representation and the individual feature representations. (2) The addition of L_{neg} also enhances the model's performance. When L_{neg} is included, accuracy improves from 0.798 to 0.814, precision from 0.780 to 0.812, and sensitivity from 0.794 to 0.820. This indicates the benefit of minimizing the similarity between different feature representations to improve overall performance. (3) Combining all three components (L_{pos}, L_{neg}, and L_{cross}) yields the best results. The model achieves the highest accuracy of 0.862, precision of 0.866, and sensitivity of 0.859 when all components are included. This confirms that the proposed contrastive learning framework, which

incorporates both positive and negative sample pair losses along with the cross-entropy loss, is the most effective configuration for optimizing performance.

4.4. The impact of different maximum sentence lengths

In an effort to understand how the maximum lengths of text in various datasets affect the performance of machine learning models, a series of ablation studies were conducted on the MFFMP-ETC. These studies were meticulously carried out across three distinct datasets to provide a comprehensive insight into the influence of text length on model outcomes. The findings, as presented in Fig. 2, reveal an intriguing pattern. Initially, the performance of the model was observed to improve consistently across all three datasets as the maximum length of the text was incrementally increased. This improvement suggests that longer texts, up to a certain point, enable the model to capture more contextual information, which contributes to more accurate predictions. However, this trend of improvement does not persist beyond a certain text length. The data indicates that when the length of the text surpasses 150 tokens, the performance gains of the model are curtailed, and a slight decline in performance is noted. This could be due to the model's capacity to process information reaching saturation, or the inclusion of noise or irrelevant information that may dilute the signal-to-noise ratio. Recognizing this inflection point, it was concluded that extending the text length beyond 150 tokens does not yield significant performance improvements and may even be detrimental. This insight is crucial for optimizing the model's efficiency and effectiveness. A maximum text length of 150 tokens was selected as the optimal value, striking a balance that maximizes the model's predictive power while minimizing computational expenses.

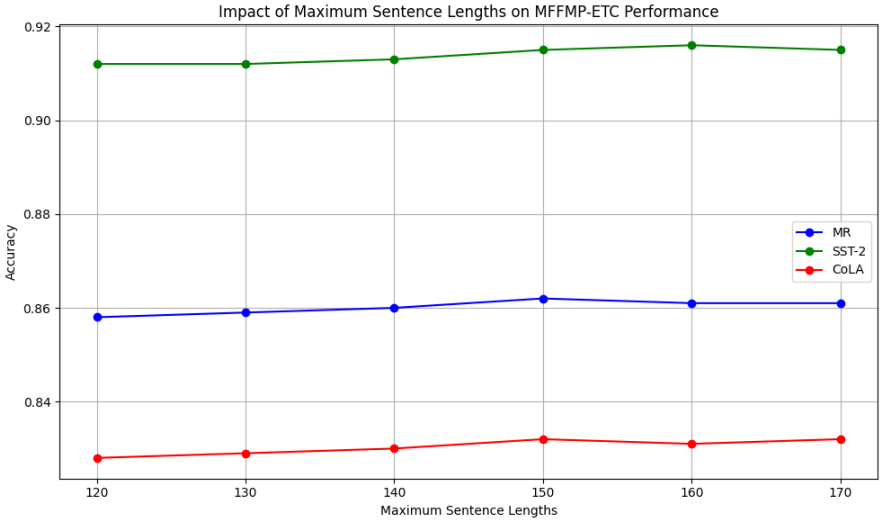


Fig. 2. The impact of different maximum sentence lengths on the performance of MFFMP-ETC

4.5. Parameter Analysis

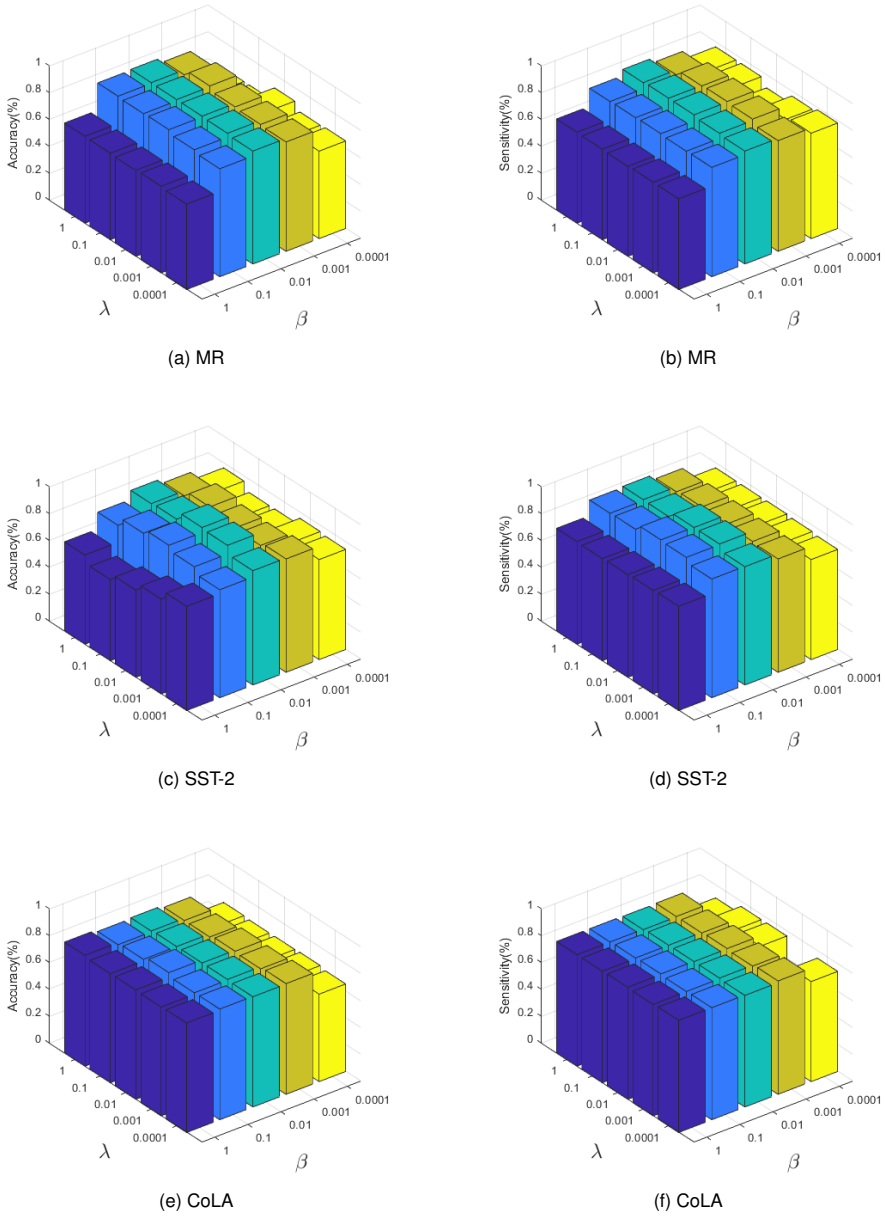
Fig. 3 illustrates the sensitivity analysis of parameters λ and β on three datasets for MFFMP-ETC. Specifically, in the experiments, λ and β are constrained within the set $\{0, 0.0001, 0.001, 0.01, 0.1, 1\}$ where one parameter was kept constant while the other was systematically varied. The results depicted in Fig. 3 illustrate MFFMP-ETC's robustness across varying values of λ and β. The performance consistently remains satisfactory, particularly when λ is set to 0.01 and β is set to 0.001. As a result, for the three datasets, MFFMP-ETC is configured with $\lambda = 0.01$ and $\beta = 0.001$ in the experiments. This configuration is empirically determined to yield a high level of accuracy, suggesting that it effectively weights the different components of the network loss function in a manner that is conducive to learning robust action recognition features. The parameters λ and β play a crucial role in the network by controlling the balance between certain regularization terms and the overall loss. The optimal values provide a valuable reference for future research and applications of MFFMP-ETC, as they offer a blueprint for achieving high performance with a reasonable computational cost.

4.6. Comparison and analysis of different feature fusion

To validate the effectiveness of the multi-feature fusion model in English text classification, MFFMP-ETC designs three types of feature fusions: (1) fusion of features extracted by multilingual BERT and BiLSTM, (2) fusion of features extracted by multilingual BERT and TEXTCNN, and (3) fusion of features extracted by TEXTCNN and BiLSTM. Based on the results in Table 4, three key observations can be made. First, the fusion of multilingual BERT and TEXTCNN demonstrates higher accuracy, precision, and sensitivity (0.850, 0.854, and 0.852, respectively) compared to the other two feature fusion methods involving multilingual BERT and BiLSTM or TEXTCNN and BiLSTM. Second, while the TEXTCNN and BiLSTM fusion performs slightly better than the multilingual BERT and BiLSTM fusion in terms of precision (0.844 vs. 0.845) and sensitivity (0.847 vs. 0.845), it has a marginally lower accuracy (0.841 vs. 0.847). Third, MFFMP-ETC outperforms all other methods across all metrics, achieving the highest accuracy (0.862), precision (0.866), and sensitivity (0.859), indicating its superior effectiveness in feature fusion for English text classification.

Table 4. Comparison and analysis of different feature fusion on MR

Fusion	Accuracy	Precision	Sensitivity
multilingual BERT and BiLSTM	0.847	0.845	0.845
multilingual BERT and TEXTCNN	0.850	0.854	0.852
TEXTCNN and BiLSTM	0.841	0.844	0.847
MFFMP-ETC	**0.862**	**0.866**	**0.859**



(a) MR (b) MR

(c) SST-2 (d) SST-2

(e) CoLA (f) CoLA

Fig. 3. The sensitivity analysis of parameters λ and β on three datasets for MFFMP-ETC

5. Conclusion

The paper introduces a novel English text classification model known as Multilingual Pre-trained based Multi-feature Fusion Model (MFFMP-ETC), which represents a significant advancement in natural language processing. By integrating the strengths of Multilingual BERT, Bi-LSTM, and TextCNN, MFFMP-ETC effectively captures both local and global contextual structure information in texts. Its innovative approach to feature fusion and the use of a multilingual pre-trained language model are crucial for enhancing the recognition of long-distance dependencies and contextual information. MFFMP-ETC achieves state-of-the-art results on the MR, SST-2, and CoLA datasets, with accuracies of 86.2%, 91.5%, and 83.2%, respectively, highlighting its superior accuracy and robustness in managing complex semantic structures and improving classification precision. Future work could further expand by exploring other multilingual pre-trained models, integrating additional contextualized features, handling multimodal data, conducting real-world application tests, improving scalability and efficiency, and enhancing model interpretability. These directions promise to push the boundaries of text classification technology and address a broader range of linguistic and contextual challenges.

References

1. Chen, J., Yang, Z., Yang, D.: Mixtext: Linguistically-informed interpolation of hidden space for semi-supervised text classification. arXiv preprint arXiv:2004.12239 (2020)
2. Chen, Y.: Convolutional neural network for sentence classification (2015)
3. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. Advances in neural information processing systems 29 (2016)
4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
5. Gao, J., Li, P., Laghari, A.A., Srivastava, G., Gadekallu, T.R., Abbas, S., Zhang, J.: Incomplete multiview clustering via semidiscrete optimal transport for multimedia data mining in iot. ACM Transactions on Multimedia Computing, Communications and Applications 20(6), 1–20 (2024)
6. Gao, J., Liu, M., Li, P., Laghari, A.A., Javed, A.R., Victor, N., Gadekallu, T.R.: Deep incomplete multi-view clustering via information bottleneck for pattern mining of data in extreme-environment iot. IEEE Internet of Things Journal (2023)
7. Gao, J., Liu, M., Li, P., Zhang, J., Chen, Z.: Deep multiview adaptive clustering with semantic invariance. Transactions on Neural Networks and Learning Systems (2023)
8. Gururangan, S., Dang, T., Card, D., Smith, N.A.: Variational pretraining for semi-supervised text classification. arXiv preprint arXiv:1906.02242 (2019)
9. Johnson, R., Zhang, T.: Deep pyramid convolutional neural networks for text categorization. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 562–570 (2017)
10. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. arXiv preprint arXiv:1607.01759 (2016)
11. Li, C., Peng, X., Peng, H., Li, J., Wang, L.: Textgtl: Graph-based transductive learning for semi-supervised text classification via structure-sensitive interpolation. In: IJCAI. pp. 2680–2686 (2021)
12. Li, P., Chen, Z., Yang, L.T., Gao, J., Zhang, Q., Deen, M.J.: An incremental deep convolutional computation model for feature learning on industrial big data. Transactions on Industrial Informatics 15(3), 1341–1349 (2018)

13. Li, P., Gao, J., Zhang, J., Jin, S., Chen, Z.: Deep reinforcement clustering. Transactions on Multimedia (2022)
14. Li, P., Laghari, A.A., Rashid, M., Gao, J., Gadekallu, T.R., Javed, A.R., Yin, S.: A deep multimodal adversarial cycle-consistent network for smart enterprise system. IEEE Transactions on Industrial Informatics 19(1), 693–702 (2022)
15. Lin, Y., Meng, Y., Sun, X., Han, Q., Kuang, K., Li, J., Wu, F.B.: Transductive text classification by combining gcn and bert. arxiv 2021. arXiv preprint arXiv:2105.05727
16. Liu, C., Wang, X.: Quality-related english text classification based on recurrent neural network. Journal of Visual Communication and Image Representation 71, 102724 (2020)
17. Liu, P., Qiu, X., Chen, X., Wu, S., Huang, X.J.: Multi-timescale long short-term memory neural network for modelling sentences and documents. In: conference on empirical methods in natural language processing. pp. 2326–2335 (2015)
18. Mundra, S., Mittal, N.: Fa-net: fused attention-based network for hindi english code-mixed offensive text classification. Social Network Analysis and Mining 12(1), 100 (2022)
19. Peng, H., Li, J., He, Y., Liu, Y., Bao, M., Wang, L., Song, Y., Yang, Q.: Large-scale hierarchical text classification with recursively regularized deep graph-cnn. In: world wide web conference. pp. 1063–1072 (2018)
20. Sachan, D.S., Zaheer, M., Salakhutdinov, R.: Revisiting lstm networks for semi-supervised text classification via mixed objective function. In: Proceedings of the aaai conference on artificial intelligence. vol. 33, pp. 6940–6948 (2019)
21. Shabestanı, S., Geçikli, M.: Machine learning use for english texts' classification (a mini-review). Osmaniye Korkut Ata Üniversitesi Fen Bilimleri Enstitüsü Dergisi 7(1), 414–423 (2024)
22. Taha, K., Yoo, P.D., Yeun, C., Taha, A.: Text classification: A review, empirical, and experimental evaluation. arXiv preprint arXiv:2401.12982 (2024)
23. Wang, G., Li, C., Wang, W., Zhang, Y., Shen, D., Zhang, X., Henao, R., Carin, L.: Joint embedding of words and labels for text classification. arXiv preprint arXiv:1805.04174 (2018)
24. Wang, X., Ji, H., Shi, C., Wang, B., Ye, Y., Cui, P., Yu, P.S.: Heterogeneous graph attention network. In: The world wide web conference. pp. 2022–2032 (2019)
25. Wang, Z., Liu, X., Yang, P., Liu, S., Wang, Z.: Cross-lingual text classification with heterogeneous graph neural network. arXiv preprint arXiv:2105.11246 (2021)
26. Xie, Q., Huang, J., Du, P., Peng, M., Nie, J.Y.: Inductive topic variational graph auto-encoder for text classification pp. 4218–4227 (2021)
27. Xu, J., Cai, Y., Wu, X., Lei, X., Huang, Q., Leung, H.f., Li, Q.: Incorporating context-relevant concepts into convolutional neural networks for short text classification. Neurocomputing 386, 42–53 (2020)
28. Yao, L., Mao, C., Luo, Y.: Graph convolutional networks for text classification. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33, pp. 7370–7377 (2019)
29. Zhang, H., Zhang, J.: Text graph transformer for document classification. In: Conference on empirical methods in natural language processing (EMNLP) (2020)
30. Zhang, X., Zhao, J., LeCun, Y.: Character-level convolutional networks for text classification. Advances in neural information processing systems 28 (2015)

Ruijuan Zhang is an outstanding scholar in the field of computational linguistics and natural language processing, with a particular focus on multilingual pre-trained models and text classification. She holds a position at the School of Foreign Languages, Zhengzhou University of Science and Technology in Zhengzhou, China. Ruijuan Zhang has a rich background in both linguistics and computer science, and has made significant contributions to the development and understanding of algorithms that enhance the accuracy and efficiency of multi-feature fusion models in English text classification tasks.

Received: June 30, 2024; Accepted: November 20, 2024.