IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2024.0429000

A multiscale interactive attention short text classification model based on BERT

LU ZHOU[1], PENG WANG[1], Huijun Zhang[1], Shengbo Wu[2] and Tao Zhang[2]
[1]Digital Silk Road Xinjiang Industry Investment Group Co., Urumqi 830000, China
[2]School of Software,Xinjiang University,Urumqi 830000, China

Corresponding authors: Huijun Zhang (e-mail: bwwang400@gmail.com) and Tao Zhang (xju_zhangtao@xju.edu.cn)

This work was supported in part by the Science and Technology Plan Project of Xinjiang Uygur Autonomous Region under Grant No.2022NC192.

ABSTRACT Text classification tasks aim to comprehend and classify text content into specific classifications. This task is crucial for interpreting unstructured text, making it a foundational task in the field of Natural Language Processing(NLP). Despite advancements in large language models, lightweight text classification via these models still demands substantial computational resources. Therefore, this paper presents a multiscale interactive attention short text classification model based on BERT, which is designed to address the short text classification problem with limited resources. A corpus containing news articles, Chinese comments, and English sentiment classifications is employed for text classification. The model uses BERT pre-trained word vectors as embedding layers, connects to a multilevel feature extraction network, and further extracts contextual features after feature fusion. The experimental results on the THUCNews, Today's headline news corpus, the SST-2 dataset, and the Touhou 38 W dataset demonstrate that our method outperforms all existing algorithms in the literature.

INDEX TERMS BERT, RNN, CNN, Multiscale interactive attention, Pre-training models

I. INTRODUCTION

CLASSIFICATION of text is an important task in the field of Natural Language Processing(NLP)[1], and short text is a very important class of data. At present, text classification methods based on traditional machine learning methods have become mature. Common machine learning classification algorithms include the plain Bayesian algorithm, K-Nearest Neighbors(KNN)[2] algorithm, and Support Vector Machine(SVM)[3, 4]algorithm, etc. which have achieved good results in text classification tasks. Still, there are also certain problems, such as not being able to represent the semantic order and semantic information well in the feature representation of text. In addition, the short length, feature sparsity, and high ambiguity pose significant obstacles to the effectiveness of mainstream text classification methods. There are data dimensionality These problems affect the classification efficiency of text to some extent. With the development of deep learning technology, neural network models such as Convolutional Neural Network(CNN)[5], Recurrent Neural Network(RNN)[6], Long Short-Term Memory(LSTM)[7] and autoencoder[8] are gradually applied in text classification tasks. In recent years, Transformer[9] has made remarkable achievements in the field of text classification. In 2018 Google proposed the BERT model[10], and the Bidirectional Encoder Representations from Transformers(BERT) pre-training model is also applied in text classification technology. In this paper, we use the BERT pre-training model as an encoder to extract the sequential and local semantic information on the feature map, and then perform feature interaction to fuse the semantic information of different scales to improve the text classification effect. In recent years generative large language models (Generative Pre-trained Transformer(GPT)[11, 12], Large Language Model Meta AI(Llama)[13, 14], etc.) have produced important research results in the field of text classification. However, large language models have the disadvantage of being difficult to fine-tune for downstream tasks because of the huge number of parameters the huge training cost, and the long training cycle. Y Dong et al.[15] were the first attempt to use Self-Interaction attention in text classification. Attention learned in the joint space of labels and words was used to weigh the textual representations obtained through self-interaction attention. Z Wang et al.[16] built a connection to match the category-related words in the document with the semantical-related subclasses of ground truth labels through the interactive double attentions from coarse to fine. Y Dong

et al. focussed on self-attentive interaction for tags and words, and Z Wang focussed on text coarseness for interactive attention. However, these studies did not consider the correlation between text order and word semantics.

We propose the Multiscale Interactive Attention-based Short Text Classification Model (MIAB) built upon BERT. Employing BERT as the backbone network, we extract spatial and temporal multiscale features from the feature maps generated by BERT. Furthermore, we utilize interactive attention mechanisms to facilitate feature interaction and fusion, thereby enhancing the performance of the model. In contrast to traditional self-attention mechanisms, we generate more effective attention weights. Compared to the monolithic nature of spatial or temporal features, our approach can more effectively fuse multi-scale features, thus strengthening the weighting dependence of sequences and keyword semantics.

II. RELATED WORK

Recently, the creation of short texts has emerged as a focal point of study across multiple fields[17], significantly influencing numerous practical uses, such as sentiment analysis[18], text generation[19], personalized recommendation[20], and dialog systems[21]. In particular, the process of short text classification involves the accurate identification of brief texts, including online news, search excerpts, and product critiques.

Because deep neural networks can learn more complex and higher-level feature representations, they have demonstrated impressive performance in brief text classification tasks in recent years. Kim, for example, introduced the TextCNN model, which trains a CNN with a single layer on top of pre-trained word vectors to obtain remarkable results in sentence-level classification tasks[22]. The word vectors remained the same, and the model only modified and learned the parameters of the one-layer CNN. For semisupervised short text classification, Hu et al. suggested a dual-level attention mechanism based on a heterogeneous graph attention network. The method selects representations of brief texts and additional data[23]. To generate neighbors of short text and explain the classification judgments for short texts, Lampridis et al. created a variational autoencoder that encodes text and decodes features[24]. Chen et al. proposed a knowledge-enhanced deep neural network for the classification of Chinese short texts. Two separate attention networks are used for knowledge encoding, and an RNN is initially added to collect contextual information of the sequence[25]. Recently, fine-tuned pre-trained language models (PLMs), such as BERT, Robustly optimized BERT approach(RoBERTa)[26], Text-To-Text Transfer Transformer(T5)[27] and GPT, have been developed into useful tools for using the abundance of information in NLP jobs. By fine-tuning PLMs to certain downstream tasks, potential information may be learned, and these models have demonstrated marked effectiveness in a range of NLP tasks, such as question answering[28], text classification[29] and lexical simplification[30]. Given the excellent performance of the fine-tuned PLM approach, training new models from scratch is generally accepted. For example, Reimers et al. modified a pre-trained BERT network with conjunctive and triplet structures to learn semantically meaningful embedded sentences and compare them to a similarity measure[31]. Sun et al. conducted several experiments to investigate various approaches to BERT fine-tuning. They obtained the best performance in a short text classification task[32]. Hu et al. proposed a BERT-based method to learn mental features for short text classification. This method integrates all features at the language level with the text context[33]. One approach that has been suggested is to use embedded encoders in PLMs to teach models how to understand contextual word representations. Fine-tuning methods with additional classifiers have been widely applied to elicit and utilize the rich knowledge contained in PLMs to adapt them to various NLP tasks. These methods have achieved impressive performance in a variety of downstream tasks, including short-text classification[34].

III. MODEL

The input is text utterance, the semantic information of the text pair is fully extracted by BERT, the natural language sequence is mapped into vector expression, the overall pre-training weight parameters of the dataset are obtained through the pre-training task of BERT, and then the training dataset is passed into the model to perform the sentence pair classification fine-tuning task further so that the pre-training parameters obtained in the pre-training task stage are further adapted to the dataset. Here, the idea of the RCNN is borrowed. For the text vector output from BERT, long short-term memory (LSTM) and the TEXTCNN are used further to extract the text sequential features and local semantic features, respectively. The semantic feature maps and local semantic feature maps are subjected to interactive attention operations. Finally, the generated feature maps are contacted, and multilevel text features are extracted via the MaxPool pooling layer and finally classified and output. A structural diagram of the network model is shown in Figure 1.

A. WORD EMBEDDING LAYER

Token embedding: Each word is converted into a fixed-dimensional vector. In BERT, each word is converted into a 768-dimensional vector representation. In the actual code implementation, the input text is tokenized before it is fed into the token embedding layer. In addition, two special tokens are inserted at the beginning ([CLS]) and the end ([SEP]) of the tokenization result Segment embedding: This is used to distinguish which sentence in a sentence pair a token belongs to. The segment embedding layer has only two vector representations. The first vector assigns a 0 to each token in the first sentence, and the second vector assigns a 1 to each token in the second sentence. If the input is only one sentence, then its segment embedding is zero. Position embedding: Transformers cannot encode the sequentiality of the input sequence, so they learn a vector representation at each position to encode information about the sequence order. Adding position embeddings allows BERT to understand the

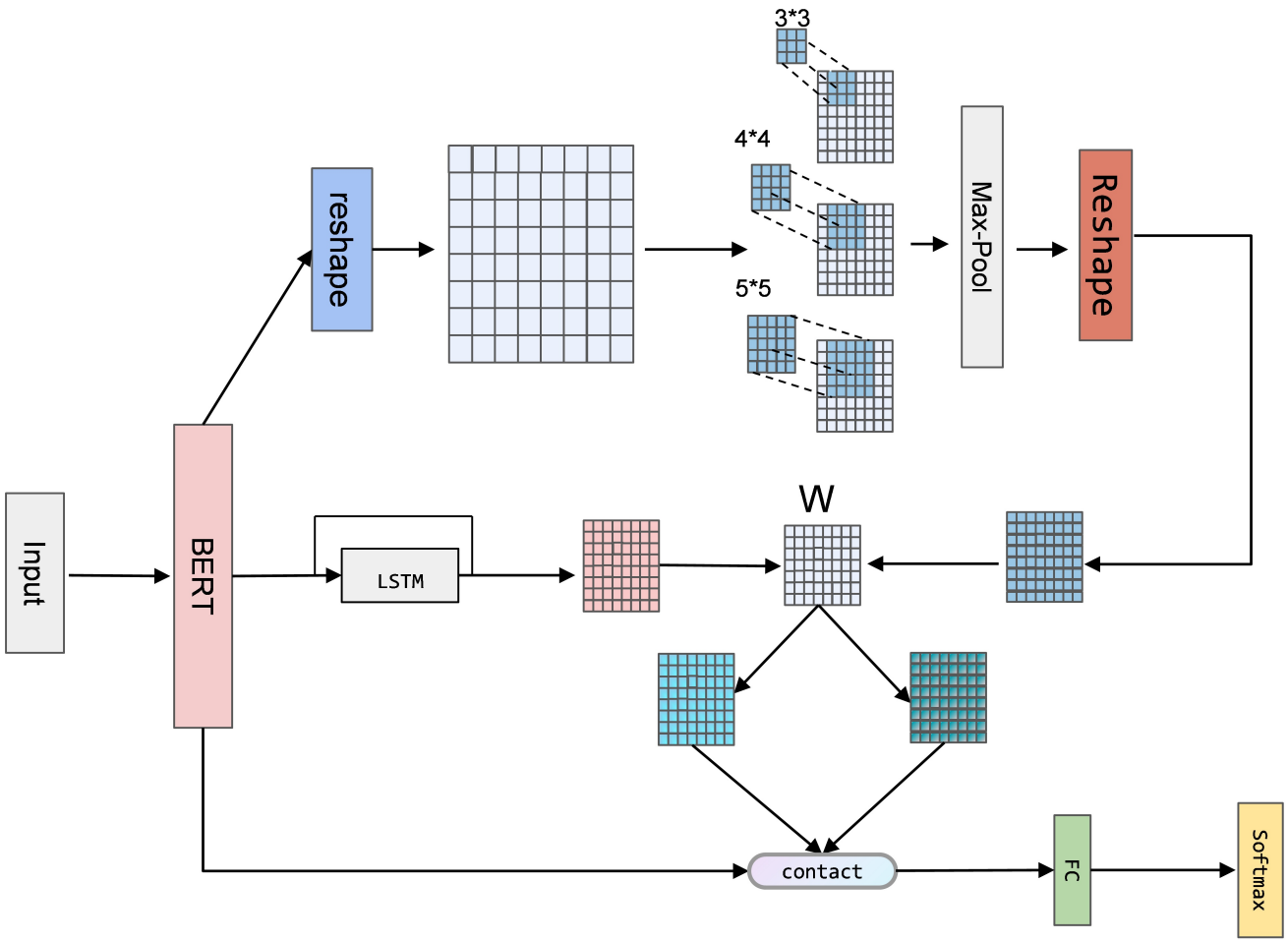


FIGURE 1. Structural diagram of the network model

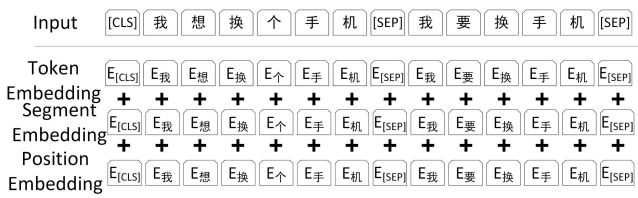


FIGURE 2. Structural diagram of BERT

following case: "I think, therefore I am", where the first "I" and the second "I" should have different vector representations. These 3 embeddings are all 768 dimensional, and in the end, they have to be summed up element by element to obtain the final 768-dimensional vector representation of each token. The word embedding layer is shown in Figure 2.

B. BERT

Google proposed the BERT pre-training model in 2018, and BERT performed well in 11 NLP task tests. BERT is based on transformer implementation. BERT contains many transformer modules, and a key factor for its success is the powerful role of the transformer. The BERT structure diagram is shown in Figure 3. The specific mathematical expressions are shown in Equation (1).

$$H_{cls} = BERT(E) \tag{1}$$

where E represents the embedding. where H_{CLS} is the representation vector of the semantic features output by BERT. The feature matrices extracted by the self-attention and spatial attention modules are added and fused, and the formula is shown below.

$$Q = W_q M \tag{2}$$

$$K = W_k M \tag{3}$$

$$V = W_v M \tag{4}$$

$$M_{att} = SelfAttention(Q, K, V) \tag{5}$$

where M represents the input embedding matrix. W_q, W_k and W_v represent three different linear layers.$SelfAttention(\bullet)$ refers to the self-attention mechanism, and $M \in \mathbb{R}^{s\_l \times h \times (s\_l/h)}$ represents the output matrix of the word embedding layer. where h is the number of attention heads.

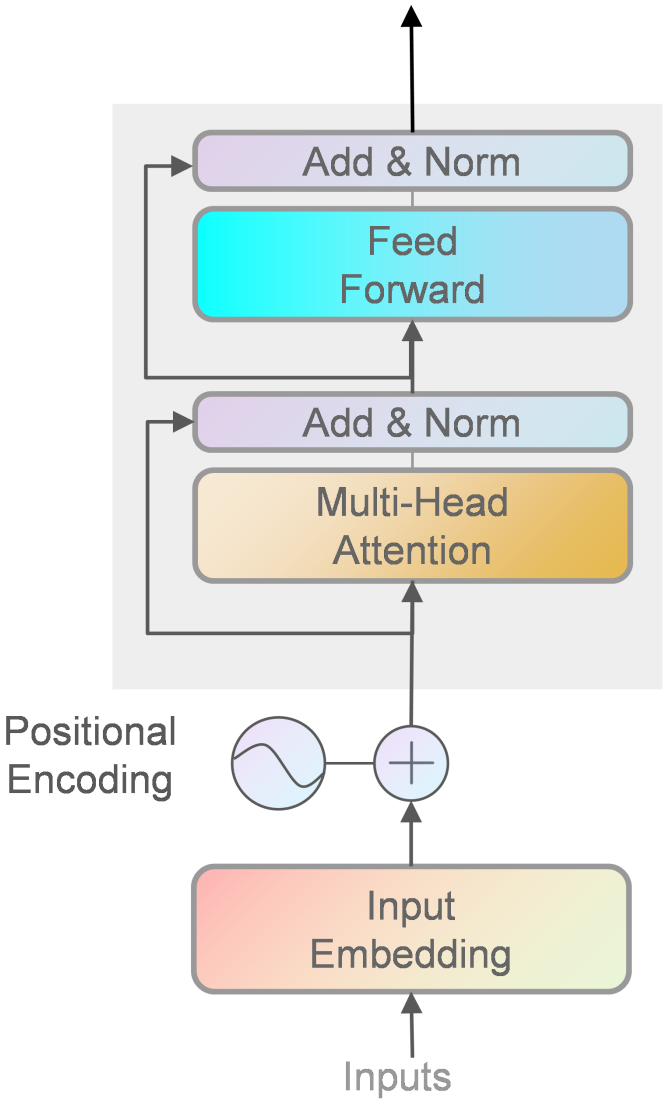


FIGURE 3. Structural diagram of BERT

As shown in Figure 4, the input embedding matrix M is linearly transformed three times via W_q, W_k, and W_v to obtain the Q, K, and V matrices and records the Q matrix as the query matrix, the K matrix as the key matrix, and the V matrix as the matrix of values. $SelfAttention(\bullet)$ is described in Eq. (6).

$$SelfAttention(Q, K, V) = softmax(\frac{Q \cdot K^T}{\sqrt{d_k}})V \tag{6}$$

C. CONVOLUTION MODULE

The feature map extracted by BERT is subjected to a convolution operation, three convolution kernels are used for the text, and the size of the convolution kernel is [3,4,5]. The output feature map is contacted, and then max pooling is performed and finally reshaped to the same size as the feature evidence output by the LSTM. The BERT structure diagram is shown in Figure 5. The specific mathematical expressions are shown in Equations (7), (8) and (9).

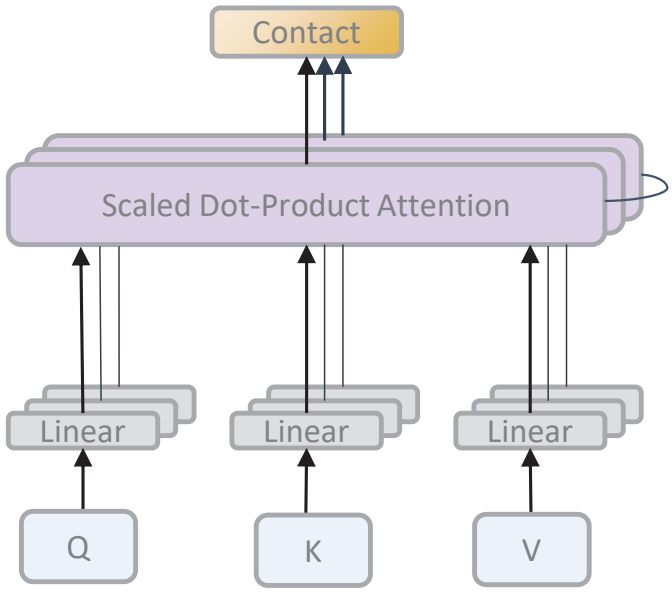


FIGURE 4. The structure of the attention combination module

$$M_{cls} = R(H_{cls}) \tag{7}$$

$$M_{conv} = Contact(Conv_{3\times3}(M_{cls}), Conv_{4\times4}(M_{cls}), Conv_{5\times5}(M_{cls})) \tag{8}$$

$$M_{conv} = MaxPooling(\sigma(M_{conv})) \tag{9}$$

where $R(\bullet)$ represents the reshape operation on the input semantic features. where M_{conv} is the semantic feature extracted after convolution, $Conv$ represents the convolution calculation, $MaxPooling$ represents the pooling operation,σ is the RuLe activation function.

D. MULTISCALE FEATURE INTERACTION MODULE

The multiscale feature interaction module aims to define a learnable parameter matrix to semantic feature information at both scales and achieve the purpose of mutual interaction between the two types of feature information. The specific process initializes the parametric evidence W and divides the matrix M1 with the convolution operation and the feature matrix M2 with the extracted temporal information for the dot product to obtain the respective interaction matrix. As the model iterates, the learning matrix learns the interaction information of the two-scale features. Finally, the two interaction matrices are in contact and then connected to the fully connected layer after the Softmax classification output. The module structure diagram is shown in Figure 6. The specific mathematical expressions are shown in Equations (10) to (16).

$$M_{Conv}{}' = M_{Conv} \odot M \tag{10}$$

$$M_{Conv}{}' = Drop(M_{Conv}{}') \tag{11}$$

$$M_{Conv}{}' = R(M_{Conv}{}') \tag{12}$$

$$M_{Lstm}{}' = M_{Lstm} \odot M \tag{13}$$

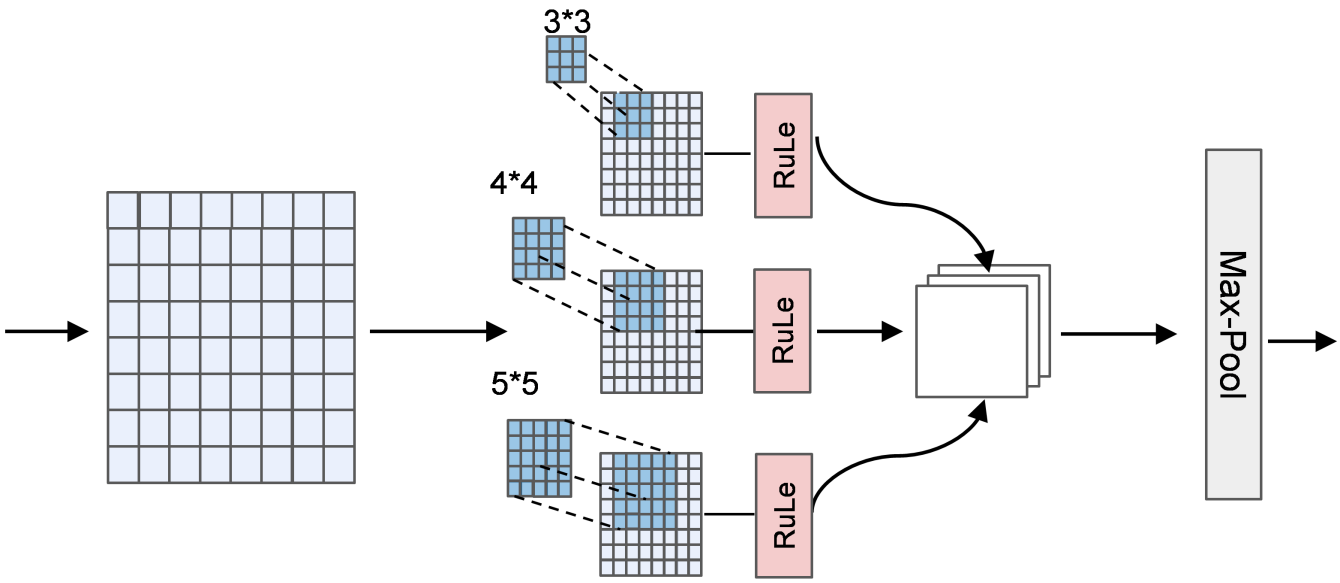


FIGURE 5. Structural diagram of the convolution module

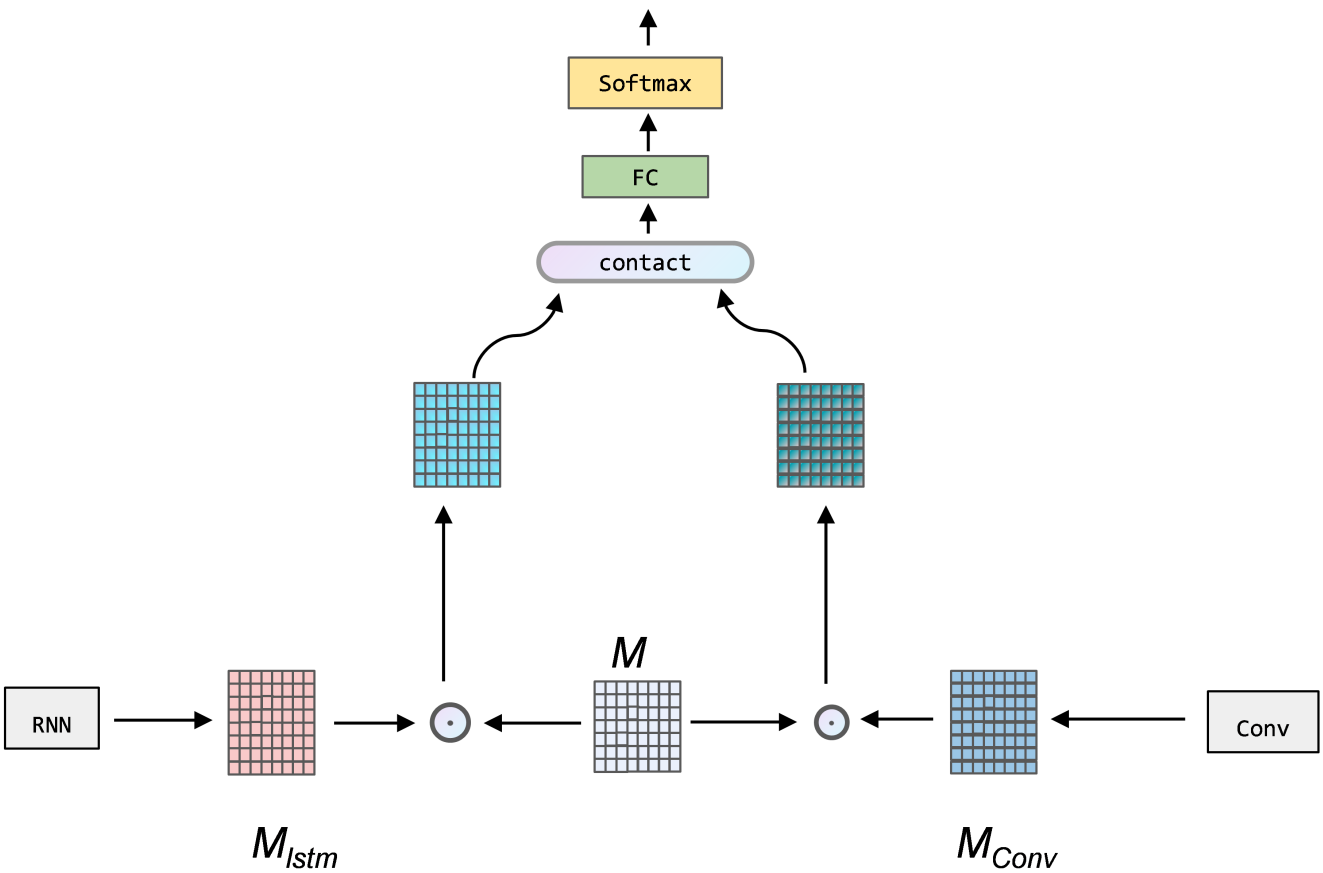


FIGURE 6. Structural diagram of BERT

$$M_{Lstm}{}' = Drop(M_{Lstm}{}') \tag{14}$$

$$Z = Contact(M_{Conv}{}', M_{Lstm}{}') \tag{15}$$

$$y = softmax(W \cdot Drop(Z) + b) \tag{16}$$

where $\odot$ represents the dot product operation and where

Drop(•) represents the Dropout operation.*Contact* represents the contact operation.W is the weight parameter, and b is the bias term.

E. LOSS FUNCTION AND EVALUATION INDEX

Loss function: We use the cross-entropy loss function to measure the deviation of the predicted value from the actual value. The cross-entropy loss function is shown in Eq. (17).

$$L = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] \tag{17}$$

where $\hat{y}$ is the predicted probability, y is the true label, and N is the number of sample classifications. The network parameters are updated according to the loss function. In this paper, precision, recall, F1, and accuracy are used as the evaluation indices of the model, and their calculation formulas are as follows:

Precision refers to the proportion of samples predicted to be positive by the model that is actually positive as a percentage of the samples predicted to be positive and is calculated via the formula shown in the following equation.

$$precision = \frac{TP}{TP + FP} \tag{18}$$

Recall refers to the proportion of actual positive samples that are also predicted to be positive to the actual positive samples and is calculated as shown in the following equation.

$$recall = \frac{TP}{TP + FN} \tag{19}$$

Accuracy is the number of correctly classified samples to the total number of samples, and the calculation formula is shown in the following equation.

$$accuracy = \frac{TP + FN}{TP + TN + FP + FN} \tag{20}$$

The F1 value is the weighted average of precision and recall and is calculated via the following equation.

$$f1 = \frac{2 * precision * recall}{precision + recall} \tag{21}$$

When the samples are unbalanced, we want to assign certain classifications different weights depending on the number of samples in each classification. This value is w_f1, and Equation (22) provides the algorithm for calculating it.

$$w\_f1 = \frac{1}{total}\sum_{k=1}^{N} f1_i * N_i \tag{22}$$

IV. EXPERIMENT AND RESULTS ANALYSIS

A. DATASET

We performed tests on the CHNSenticorp and SST-2[35] datasets, both of which are sentiment classification datasets composed of reviews, to objectively assess the methodology presented in this study. To further extend our evaluation and validate the effectiveness of our proposed module, we incorporate the THUCNew and Toutiao38w datasets into the ablation experiments. The THUCNews dataset covers several news topics, including politics, economics, technology, and society, with many news texts under each topic.

TABLE 1. Dataset

Dataset	Training Set Size	Validation Set Size	Testing Set Size
ChnSentiCorp	9146	1200	1200
SST-2	60,000	7349	872
THUCNews	192000	36000	12000
Toutiao38w	229612	76538	76538

B. EXPERIMENTAL PARAMETER

Hardware: ubuntu 16.4 system, GPU: Nvidia V100. The experiments used the PyTorch deep learning framework with an embedding dimension of 768 for each word, a total word list size of 21128, a sentence length of 32, 12 encoder layers, 12 attention matrix heads, and a BERT matrix initialization range of 0.02. The model uses the cross-entropy function as the loss function, and the AdamW optimizer is used to update the parameters. The number of epochs of the training phase is 50, and the number of epochs of the fine-tuning phase is 3. The batch size is set to 64, the learning rate is set to 2e-5, the gradient clipping max grad norm is set to 10, and the early termination batch size is set to 1000.

C. EXPERIMENTAL RESULTS

TABLE 2. ACC of the test set

model	CHNSenticorp	SST-2
TEXTCNN	84.0	78.4
FastText	84.75	71.06
DPCNN	82.83	78.83
BERT	93.24	84.95
Albert-base	92.58	86.22
ELECTRA-base	93.5	-
Roberta	92.58	87.68
Macbert	93.5	-
MIAB	**93.97**	**87.73**

TABLE 3. WF1 of the test set

model	CHNSenticorp	SST-2
TEXTCNN	84.3	78.06
FastText	84.75	70.63
DPCNN	82.75	78.42
BERT	93.66	84.88
Albert-base	92.58	86.37
ELECTRA-base	93.48	-
Roberta	92.58	87.68
Macbert	92.58	-
MIAB	**93.89**	87.62

To verify the effectiveness of the MIAB model, we conducted a comparison experiment between MIAB and the baseline, and the experimental results are shown in Table 3. The test set accuracy and WF1 value of MIAB on CHNSenticorp reached 93.97 and 93.89, respectively, which are higher than those of all the baselines; the test set accuracy and WF1 value of MIAB on SST-2 reached 87.73 and 87.62, which

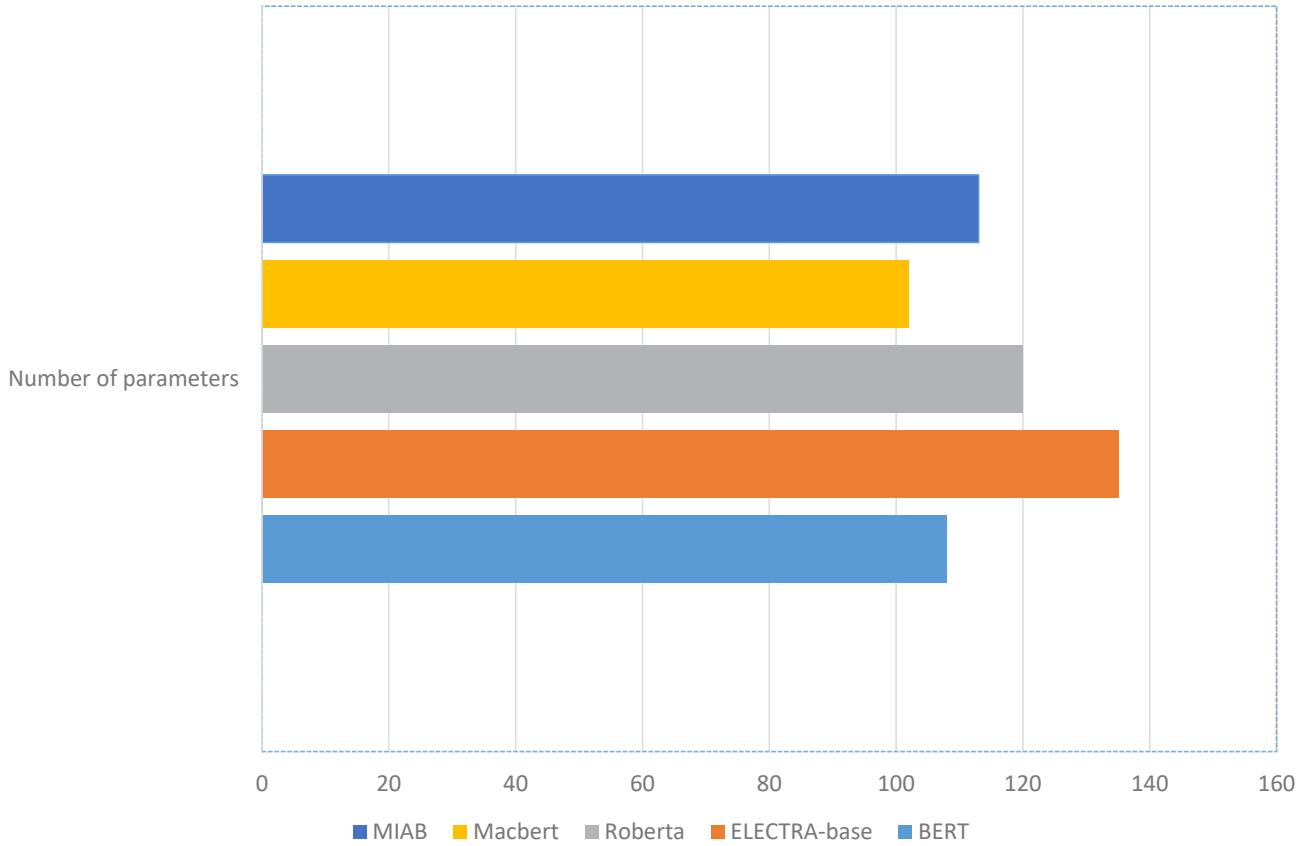


FIGURE 7. **Number of parameters**

are higher than those of all the baselines; and the accuracy and WF1 value of MIAB reached the same performance as those of Roberta. Roberta's performance. MIAB needs only 113M parameters to achieve performance in line with the best RoBERTa baseline. In addition, Roberta needs 120M parameters. The experimental results indicate that MIAB has better performance on the Chinese dataset. A comparison of the number of parameters of the MIAB model with that of the baseline model is shown in Figure 7. MIAB has an advantage over mainstream pre-trained models in terms of the number of parameters metric.

D. ABLATION EXPERIMENT

We perform ablation experiments on MIAB, and the results are shown in Table 4. Through comparative experiments, we find that MIAB has better performance on Chinese datasets, and we conduct ablation experiments on two Chinese datasets, THUCNews and Toutiao38w, to verify the effectiveness of MIAB. where BERT-BILSTM stands for yes replacing the multiscale feature interaction module with BILSTM. The test set accuracy and F1 value of MIAB on THUCNews reach 94.86 and 94.86, respectively, which are higher than those of BERT (93.83 and 93.82, respectively). Table 3 shows that the performance of MIAB on Toutiao38w is close to that of BERT, but both are higher than that of TEXTCNN. TEXTCNN scores lower in accuracy and recall, and the overall classification performance ability is poor because TEXTCNN can obtain local information ability very well but obtain global semantic feature ability. BERT can extract text features and directly use the classifier for classification, and the performance on the dataset is greater than that of TextCNN. Access to the RNN after BERT can extract text temporal information, but the single-layer RNN has a limited ability to extract features compared with BERT direct classification, which does not significantly improve performance. When the output of BERT uses a multilayer RNN combined with a self-attention mechanism to obtain temporal features to improve the lack of temporal information acquisition ability of BERT, the experiment proved to be effective. The experimental data show that the MIAB model has improved in all the indices compared with BERT, and it is also better than other comparative experimental models, which proves the effectiveness of the MIAB model. As shown in Table 4, MIAB increases the number of model parameters due to the addition of the multilayer self-attention recurrent network module, resulting in a 22.7% increase in the model convergence time compared with BERT and a 5.5% increase in the convergence time compared with BERT-LSTM, which is also a disadvantage of the model, as it currently exists.

We performed ablation experiments for each convolu-

tional layer, namely, no convolutional layer (0 convolutional blocks), only convolutional blocks of dimensions 3×3, only two convolutional blocks of dimensions 3×3 and 4×4, and three convolutional blocks of dimensions 3×3, 4×4 and 5×5. The experiments in Table 6 show that the best performance is achieved when the model contains 3×3, 4×4, and 5×5 convolutional blocks. The analysis reveals that three different scales of convolutional blocks extract greatly enriched spatial semantic features. Through Figure 8, we find that the use of a single convolutional block leads to a degradation in the performance of the model, and through analysis, it is found that a single convolutional block extracts a limited number of features that will overlap with the features in the LSTM network, leading to a degradation in performance.

TABLE 4. ACC of the test set

model	THUCNews	Toutiao38w
TEXTCNN	90.08	84.61
BRT+LSTM	93.98	87.81
BERT	93.57	87.83
MIAB	94.86	87.62

TABLE 5. WF1 of the test set

model	THUCNews	Toutiao38w
TEXTCNN	90.06	78.3
BRT+LSTM	93.27	86.7
BERT	93.57	86.8
MIAB	94.86	87.70

TABLE 6. One epoch iteration time

model	THUCNews	SST-2	Toutiao38w
BERT	18m09s	24m24s	938s
BRT+LSTM	20m27s	29m00s	1099s
BRT+CNN	22m46s	233m73s	1132s
MIAB	36m25s	37m00s	1151s

V. VISUAL ANALYSIS

In this section, Figure 9 shows the heatmap of the attention score in the transformer, and the heatmap of the interaction attention matrix in MIAB is shown visually. Taking the Chinese example sentence [the quality of this product is truly too bad], which is a slightly negative sentiment output, the theoretically analyzed model's attentional weights should be distributed mainly among the keywords that express negative sentiment. The distribution of attention in the interaction attention matrix is labeled in Figures 10 and 11. Figures 10 and 11 show that the interaction attention score enhances the relevance of words at different positions in the text while preserving the self-attention weights in the transformer; this proves that the interactive attention matrix is effective in fusing semantic features at different scales, thus making the model more focused on the parts that characterize sentence classification.

VI. CONCLUSIONS

Compared with the baselines, the MIAB proposed in this paper significantly improves the accuracy and F1 value. The multiscale feature interaction module can fully integrate the semantic temporal features and local spatial features of the text, thus realizing a significant performance improvement on the basis of BERT. Moreover, due to the introduction of additional modules, the model has an increased number of parameters, and the downstream task fine-tuning time is increased. In the next step of the research, the model will be validated for long text categorization, as well as for uniting more features of different scales, such as coarse-graining of text and different lexical properties. Lightweight research on pre-trained models that use existing network structures to improve the computational efficiency of the models is also a future research consideration.

REFERENCES

[1] Guangquan Lu, Jiangzhang Gan, Jian Yin, Zhiping Luo, Bo Li, and Xishun Zhao. Multi-task learning using a hybrid representation for text classification. *Neural Computing and Applications*, 32(11):6467–6480, 2020.

[2] Gongde Guo, Hui Wang, David Bell, Yaxin Bi, and Kieran Greer. Knn model-based approach in classification. In *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy, November 3-7, 2003. Proceedings*, pages 986–996. Springer, 2003.

[3] Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28, 1998.

[4] Huiyan Li. Text recognition and classification of english teaching content based on svm. *Journal of Intelligent & Fuzzy Systems*, 39(2):1757–1767, 2020.

[5] Yahui Chen. Convolutional neural network for sentence classification. Master's thesis, University of Waterloo, 2015.

[6] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. Recurrent neural network for text classification with multi-task learning. *arXiv preprint arXiv:1605.05101*, 2016.

[7] Gang Liu and Jiabao Guo. Bidirectional lstm with attention mechanism and convolutional layer for text classification. *Neurocomputing*, 337:325–338, 2019.

[8] Weidi Xu, Haoze Sun, Chao Deng, and Ying Tan. Variational autoencoder for semi-supervised text classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirec-

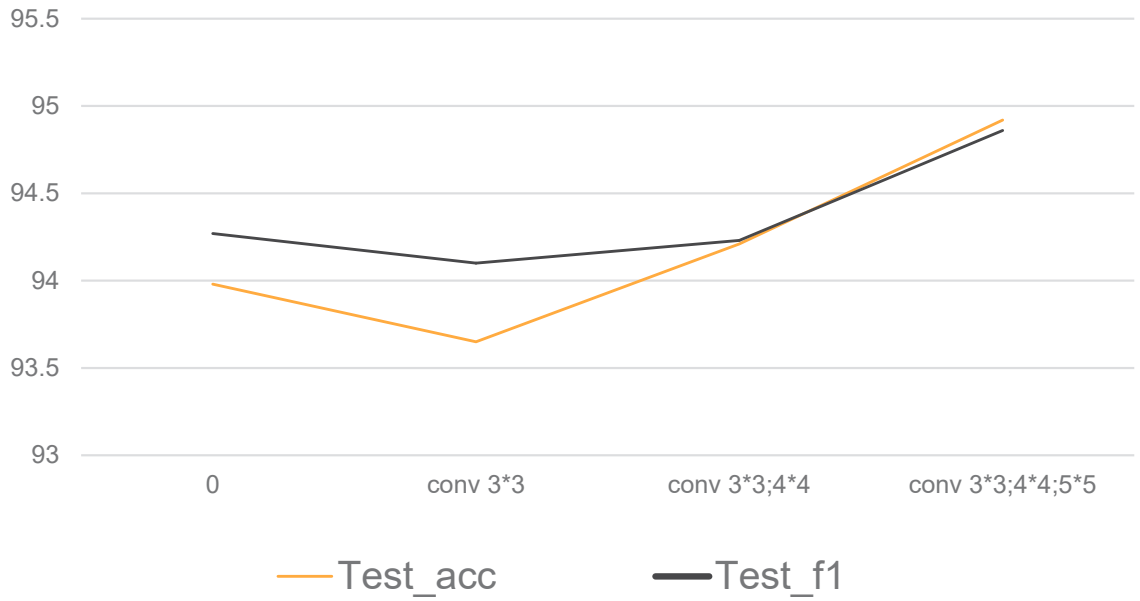


FIGURE 8. **The structural diagram of BERT**

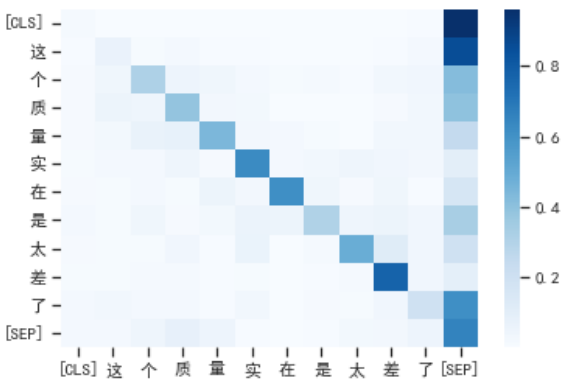


FIGURE 9. **Transformer self-attention score visualization**

tional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[11] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[12] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[13] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[14] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[15] Yanru Dong, Peiyu Liu, Zhenfang Zhu, Qicai Wang, and Qiuyue Zhang. A fusion model-based label embedding and self-interaction attention for text classification. *IEEE Access*, 8:30548–30559, 2019.

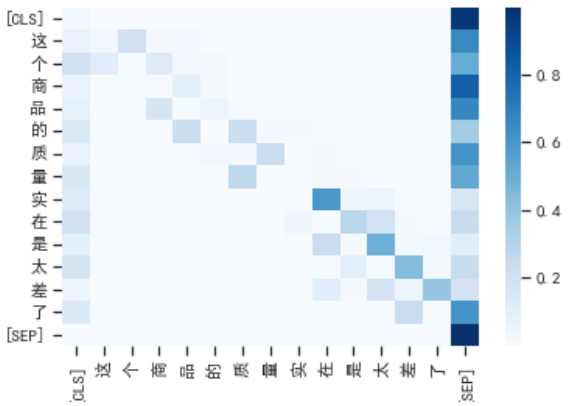


FIGURE 10. **Interaction attention score visualization 1 (length=64)**

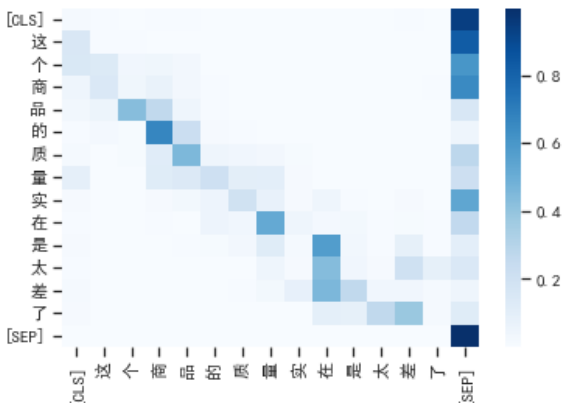


FIGURE 11. **Interaction attention score visualization 2 (length=128)**

[16] Ziyuan Wang, Hailiang Huang, and Songqiao Han. Idea: Interactive double attentions from label embedding for text classification. In *2022 IEEE 34th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 233–238. IEEE, 2022.

[17] Jipeng Qiang, Zhenyu Qian, Yun Li, Yunhao Yuan, and Xindong Wu. Short text topic modeling techniques, applications, and performance: a survey. *IEEE Transactions on Knowledge and Data Engineering*, 34(3):1427–1445, 2020.

[18] Chao Song, Xiao-Kang Wang, Peng-fei Cheng, Jianqiang Wang, and Lin Li. Sacpc: A framework based on probabilistic linguistic terms for short text sentiment analysis. *Knowledge-Based Systems*, 194:105572, 2020.

[19] Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, and Hannaneh Hajishirzi. Text generation from knowledge graphs with graph transformers. *arXiv preprint arXiv:1904.02342*, 2019.

[20] Kun Zhou, Wayne Xin Zhao, Shuqing Bian, Yuanhang Zhou, Ji-Rong Wen, and Jingsong Yu. Improving conversational recommender systems via knowledge graph based semantic fusion. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1006–1014, 2020.

[21] Ji Young Lee and Franck Dernoncourt. Sequential short-text classification with recurrent and convolutional neural networks. *arXiv preprint arXiv:1603.03827*, 2016.

[22] Hwa-Yeon Kim, Jinsu Lee, Na Young Yeo, Marcella Astrid, Seung-Ik Lee, and Young-Kil Kim. Cnn based sentence classification with semantic features using word clustering. In *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 484–488. IEEE, 2018.

[23] Hu Linmei, Tianchi Yang, Chuan Shi, Houye Ji, and Xiaoli Li. Heterogeneous graph attention networks for semi-supervised short text classification. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4821–4830, Hong

Kong, China, November 2019. Association for Computational Linguistics.

[24] Orestis Lampridis, Riccardo Guidotti, and Salvatore Ruggieri. Explaining sentiment classification with synthetic exemplars and counter-exemplars. In *International Conference on Discovery Science*, pages 357–373. Springer, 2020.

[25] Jindong Chen, Yizhou Hu, Jingping Liu, Yanghua Xiao, and Haiyun Jiang. Deep short text classification with knowledge powered attention. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 6252–6259, 2019.

[26] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[27] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

[28] Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, et al. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*, 2020.

[29] Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. Deep learning–based text classification: a comprehensive review. *ACM computing surveys (CSUR)*, 54(3):1–40, 2021.

[30] Jipeng Qiang, Yun Li, Yi Zhu, Yunhao Yuan, and Xindong Wu. Lexical simplification with pretrained encoders. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8649–8656, 2020.

[31] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

[32] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. How to fine-tune bert for text classification? In *Chinese computational linguistics: 18th China national conference, CCL 2019, Kunming, China, October 18–20, 2019, proceedings 18*, pages 194–206. Springer, 2019.

[33] Yongjun Hu, Jia Ding, Zixin Dou, and Huiyou Chang. Short-text classification detector: A bert-based mental approach. *Computational intelligence and Neuroscience*, 2022, 2022.

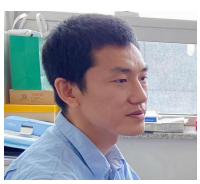
[34] Xu Han, Zhengyan Zhang, Ning Ding, Yuxian Gu, Xiao Liu, Yuqi Huo, Jiezhong Qiu, Yuan Yao, Ao Zhang, Liang Zhang, et al. Pre-trained models: Past, present and future. *AI Open*, 2:225–250, 2021.

[35] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

• • •

ZHOU LU M.S. degree in Software Engineering, Beijing Institute of Technology, 2013. He is currently working as an R&D director and Deputy Senior Engineer, with research interests in edge computing and big data processing.

PENG WANG received his B. S Degree in Computer Software from Shandong University of Petroleum in 2003 and is currently working as a senior engineer in the Digital Silk Road Xinjiang Industrial Investment Group Limited, engaging in the research direction of data governance and deep learning.



HUIJUN ZHANG received his B. S degree in Mechatronics from Southwest Petroleum University in 2016 and is currently working as an associate senior engineer at Digital Silk Road Xinjiang Industrial Investment Group Limited, with research interests in edge computing and affective computing.

SHENGBO WU He is currently working toward the M.S. degree in artificial intelligence with the Department of Xinjiang University, Urumqi, China. His research interests are in natural language processing.

TAO ZHANG Serving as an Associate Professor at Xinjiang University in 2021, his research interests are in natural language processing and multimodal recognition. E-mail: xju_zhangtao@xju.edu.cn Processing.