PeerJ Computer Science

Joint coordinate attention mechanism and instance normalization for COVID online comments text classification

Rong Zhu[1], Hua-Hui Gao[1] and Yong Wang[2]

[1] School of Computer Science, Qufu Normal University, Rizhao, China
[2] Laboratory Experimental Teaching and Equipment Management Center, Qufu Normal University, Rizhao, China

ABSTRACT

Background. The majority of extant methodologies for text classification prioritize the extraction of feature representations from texts with high degrees of distinction, a process that may result in computational inefficiencies. To address this limitation, the current study proposes a novel approach by directly leveraging label information to construct text representations. This integration aims to optimize the use of label data alongside textual content.

Methods. The methodology initiated with separate pre-processing of texts and labels, followed by encoding through a projection layer. This research then utilized a conventional self-attention model enhanced by instance normalization (IN) and Gaussian Error Linear Unit (GELU) functions to assess emotional valences in review texts. An advanced self-attention mechanism was further developed to enable the efficient integration of text and label information. In the final stage, an adaptive label encoder was employed to extract relevant label information from the combined text-label data efficiently.

Results. Empirical evaluations demonstrate that the proposed model achieves a significant improvement in classification performance, outperforming existing methodologies. This enhancement is quantitatively evidenced by its superior micro-F1 score, indicating the efficacy of integrating label information into text classification processes. This suggests that the model not only addresses computational inefficiencies but also enhances the accuracy of text classification.

Submitted 8 May 2024
Accepted 14 July 2024
Published 19 August 2024

Corresponding author
Rong Zhu, zhurongsd@qfnu.edu.cn

Academic editor
José Alberto Benítez-Andrades

Additional Information and Declarations can be found on page 16

DOI 10.7717/peerj-cs.2240

Copyright
2024 Zhu et al.

Distributed under
Creative Commons CC-BY 4.0

OPEN ACCESS

Subjects Artificial Intelligence, Data Mining and Machine Learning, Network Science and Online Social Networks, Text Mining
Keywords Coordinate attention mechanism, Instance normalization, Text classification, Gaussian Error Linear Unit, COVID online reviews

INTRODUCTION

Text classification has been explored in various fields such as sentiment analysis (*Zainuddin, Selamat & Ibrahim, 2018*) and questions and answers (*Gweon & Schonlau, 2024*). In the age of information explosion, manually processing and categorizing large amounts of text data is both time-consuming and challenging. In addition, the accuracy of manual text categorization is easily affected by human factors. Text classification has been used for several years; however, the classification methods have focused on input text manipulation.

How to cite this article Zhu R, Gao H-H, Wang Y. 2024. Joint coordinate attention mechanism and instance normalization for COVID online comments text classification. *PeerJ Comput. Sci.* **10**:e2240 http://doi.org/10.7717/peerj-cs.2240

Traditional text classification models dominate, such as Plain Bayes (NB) (*Xu, 2018*), K-nearest neighbors (KNN) (*Alhutaish & Omar, 2015*) and support vector machines (SVM) (*Wan et al., 2012*). Later, deep neural networks such as convolutional neural networks (CNNs) (*Malik & Jain, 2024*; *Zhu, 2021*) and recursive neural networks (*Li et al., 2020*) proved more effective in text encoding. Subsequently, Bidirectional Encoder Representations from Transformers (BERT) (*Sung, Park & Kim, 2023*), XLNet (*Wu, Wang & Zhao, 2023*), and other large pre-training models achieved substantial performance improvements owing to their powerful coding capabilities. However, these methods usually rely on highly differentiated text representations and require significant computational resources.

To mitigate resource limitations, this study leverages label information to address such challenges. In single-cell research (*Qu, Kao & Hakonarson, 2024*), scientists aim to uncover the details of cellular heterogeneity and dynamic changes through high-resolution analysis of individual cells. This field has rapidly progressed due to advancements in high-throughput sequencing technologies, enabling researchers to measure and analyze gene expression in single cells. This leads to a deeper understanding of cell types, states, and functions. A key objective in single-cell research is to identify and classify different cell types, which is crucial for comprehending the composition and function of complex biological systems, such as human tissues or tumors. To achieve this, researchers have developed various algorithms and tools to extract meaningful features from single-cell RNA sequencing data for cell classification and annotation.

Single-cell research is crucial because it allows scientists to study the gene expression of individual cells, revealing cellular heterogeneity and dynamic changes that are not detectable in bulk cell analyses. This detailed understanding is vital for identifying and classifying different cell types, understanding their functions, and discovering new cell states. This is important for various aspects of biological research, including developmental biology, disease mechanisms, and regenerative medicine. In connection to RNA studies, single-cell RNA sequencing (scRNA-seq) enables precise measurement of RNA molecules in individual cells, providing insights into gene expression patterns and regulatory mechanisms at an unprecedented resolution. This helps in understanding the complexity of biological systems, identifying biomarkers, and developing targeted therapies.

Wu et al. (2023) focused on the label information between cells to better extract features, which also inspired text classification tasks. In a text classification task, the role of labels is to capture more relevant words during classification. *Wang et al. (2018)* developed an attention model known as LEAM, which integrated the label and word vectors into the same space through construction using label embeddings. *Du et al. (2019)* added an interactive mechanism to the process of text classification to enable the model to obtain the corresponding word-matching signals in the classification process.

Experiments confirmed that the above model maintained good performance under the premise of a simpler architecture and fewer parameters. However, the attention adopted by the model considers only the effects of text labels. Embedded label information was not fully used. Therefore, this study further combines the attention from text to label, integrates text and label, and makes the model look for more labels to match the text in the process of encoding label embedding. In comparison with previous label embedding methods, the fusion method of text and labels makes full use of the feedback information of text representations and encodes this feedback into labels.

Deep-learning models involving attention mechanisms (*Zhang & Wu, 2023*) have been used in classification and chromosome science. The coordinate attention mechanism (*Lu et al., 2016*; *Yu et al., 2017*; *Lu et al., 2019*) is widely used in multi-channel learning between images and languages. *Lu et al. (2016)* applied a coordinate attention model to image inference. Recently, to enhance the learning of image content, *Lu et al. (2019)* adopted a coordinate attention transformer to embed images and text. The study referenced in *Vaswani et al. (2017)* proposes a novel architecture entirely based on attention mechanisms, designed to replace traditional recurrent neural networks and convolutional neural networks. This methodology facilitates the generation of mutually attentive representations, enabling the explicit capture of relationships between text and labels. *Liu et al. (2022)* adopted the IN method within a self-attention model, subsequently applying an activation function to enhance the model's functionality. This modification not only provides stochastic regularization but also markedly augments the overall performance of the model. Our model is built on the basis of this framework.

This research introduces a classification algorithm predicated on a coordinate attention mechanism enhanced by IN (*Ulyanov, Vedaldi & Lempitsky, 2016*). The model comprises two principal components: an IN Text and Label Coordinate Attention Encoder (INTLCE) and an Adaptive Label Decoder (ALD). The INTLCE is engineered to generate interactive representations that elucidate the interconnections between texts and labels. Conversely, the ALD is designed to delineate and apprehend the correlations amongst labels. The salient contributions of this study are delineated as follows: (1) The proposed model integrates text and label information, thereby augmenting the utilization of label data through the synergistic fusion of text and labels, which underscores the model's innovative approach to considering the symbiotic relationship between text and label information. (2) Enhancing the conventional coordinate attention mechanism, this study incorporates IN and GELU functions. These additions are aimed at augmenting model efficacy while concurrently mitigating training challenges and computational exigencies. (3) The model's efficacy was assessed employing the COVID online comments dataset, wherein it demonstrated competitive performance relative to contemporaneous studies utilizing the identical dataset. This evaluation underscores the model's robustness and its potential applicability to real-world datasets.

COORDINATE ATTENTION MECHANISM MODEL WITH IN METHOD

We proposed an optimized coordinate attention network. First, we unified the encoding and embedding of labels and text. Thereafter, the IN method was used to accelerate the convergence ability of the model and reduce the complexity of training. Finally, their

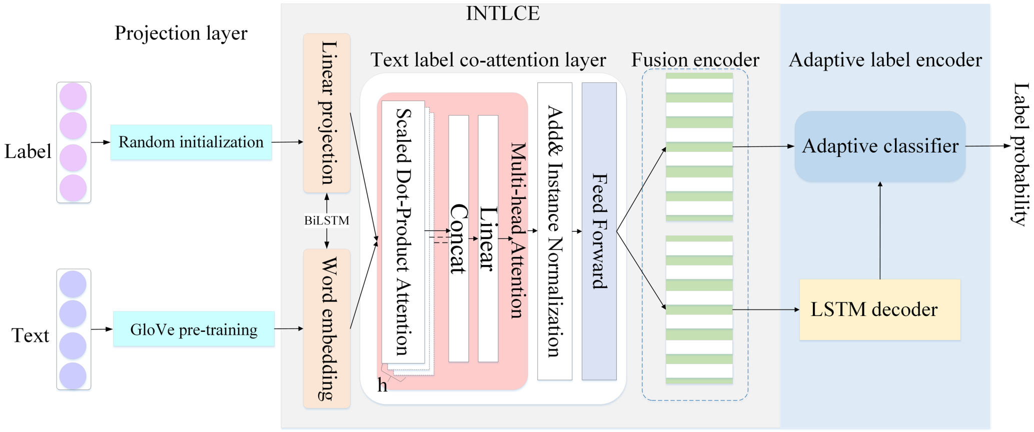

Figure 1 **Model architecture.**

Full-size DOI: 10.7717/peerjcs.2240/fig-1

common participation representations were used to generate the target labels. The structure of the model is shown in Fig. 1.

In this study, the label generation problem was applied to text classification. Specifically, given the text x , our goal was to generate probability $\hat{y}$ for all categories.

Overall plan

Let a text sequence with m words be represented as $x = [x_1, \ldots, x_m]$, and let the set of labels be $\wp = \{l_1, \ldots, l_c\}$, where c denotes the total number of categories (such as the binary classification used in this study, $c = 2$). The labels are divided into negative and positive categories, which prepare the model for learning and final validation. Note that the text sequence x in the document contains m words, and the label sequence l in document contains $\wp$ labels. The mapping process is shown in the following equation. The label are associated with the note text denoting z_{text}, and the label with the note text denoting z_{label}.

$$z_{text}, z_{label} = f_{enc}(x, l),$$

where f_{enc} represents the process of encoding a mapping function. After entering the two representations of z_{text} and z_{label} obtained using above formula, we use the decoder to generate the probability sequence $\hat{y}$ as follows:

$$\hat{y} = f_{dec}(z_{text}, z_{label}),$$

where f_{dec} represents the process of decoding the mapping function. Using the above equation, the decoder can use the mutual representation of the text and labels to make a final prediction.

INTLCE

In this section, we discuss the INTLCE module in detail. INTLCE encodes text and label sequences into mutually participating labels and text representations. Specifically, INTLCE is divided into the projection, text label coordinates attention and fusion encoder layers.

We used the bidirectional long short term memory (BiLSTM) (*Graves & Schmidhuber, 2005*) approach to effectively encode words and labels and enhance our understanding of input sequences. In addition, this approach helps identify correlations between labels, which further improves the accuracy of the encoding process.

Given a sequence of words and labels $x \in R^m$ and $l \in R^c$, we start by mapping words and labels separately to the word-embedding layer $x_{emb} \in R^{m \times d_{emb}}$ and label-embedding layer $L_{emb} \in R^{c \times d_{emb}}$ respectively, where d_{emb} denotes the embedding dimension.

For each sentence, we used pre-trained GloVe word embedding (*Kamyab, Liu & Adjeisah, 2021*; *Ibrahim et al., 2021*). This is an unsupervised word-vector representation technique. The resulting representation depicts the linear substructure in the word vector space by training the aggregated global word-word co-occurrence statistics in the corpus.

For each label, a random initialization was used for embedding. To improve computational efficiency, we first used an independent linear projection layer. Suppose x_{enb} and L_{emb} are projected separately into a more compact, smaller-dimensional embedding, where $X_{proj} \in R^{m \times d}$, $L_{proj} \in R^{c \times d}$, $d < d_{emb}$, d is a hidden dimension.

Thereafter, we used the BiLSTM method for word embedding $X_{proj} \in R^{m \times d}$ and label embedding $L_{proj} \in R^{c \times d}$ in the projection layer. The calculation equations are expressed as follows:

$$X_{enc} = BLM\left(X_{proj}\right),$$
$$L_{enc} = BLM\left(L_{proj}\right),$$

where BLM is short for BiLSTM, X_{enc} denotes text encoding and L_{enc} is the label coding. In our implementation, we incorporated weight sharing into the BiLSTM. This means that the same weights are used for both the forward and backward directions of the network, which can help reduce the number of parameters and improve efficiency.

Furthermore, the study introduces a novel normalization technique, termed IN, which represents an innovative approach to the standardization of data within the experimental framework. This method enhances the analytical rigor by ensuring consistency in data treatment, thereby improving the reliability and interpretability of the results. IN is independent of the channel or batch size and ensures the independence of each text instance, which enhances the performance of the model. Another optimization approach involves replacing the original self-attention mechanism activation function with the GELU functions. Random regularization was added to make the model more consistent with the cognitive processes.

First, the key steps in optimizing the coordinate attention mechanism focus on the scaled dot product attention, which is expressed as follows:

$$Attention(Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{d_h}}\right)V,$$

where $Q \in R^{q \times d_k}$, $K \in R^{k \times d_k}$, and $V \in R^{v \times d_v}$. The multi-head attention equation is expressed as follows:

$$MHead(Q, K, V) = Softmax\left(\left[H_1; \ldots; H_p\right]\right)W^0,$$

where $H_i = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right)$, and the projection parameters are $W_i^Q \in R^{d_k \times d_p}$, $W_i^K \in R^{d_k \times d_p}$, $W_i^V \in R^{d_v \times d_p}$, and $W_i^O \in R^{pd_p \times d_h}$. We used $d_k = d_v = d_p$ and $d_p = d_h/p$ as the dimensions of each head, where d represents the dimension of the interval model, p represents the number of heads, and $[\cdot]$ represents the join operation.

Next, because the traditional self-attention mechanism (Vaswani et al., 2017) only considers text modes, the matrices Q, K, and V represent text encodings. The text code X_{enc} and label code L_{enc} were simultaneously input into multiple attention modules, and the self-attention module was converted into a coordinate attention module.

$$X_{att} = MHead_X\left(X_{enc}, L_{enc}, L_{enc}\right),$$
$$L_{att} = MHead_L\left(L_{enc}, X_{enc}, X_{enc}\right),$$

where $X_{att} \in R^{m \times d_h}$ and $L_{att} \in R^{c \times d_h}$ denote the text representations of the label participation and text participation, respectively. The term d_h represents the hidden dimensions of the coordinate attention layer.

Furthermore, after IN, the residual connection and feedforward network (FN) obtain text and label fusion, encoding $X_{fu} \in R^{m \times d}$ and $L_{fu} \in R^{c \times d}$.

$$X_{fu} = IN_X\left(FN_X\left(X_{att}\right) + X_{enc}\right),$$
$$L_{fu} = IN_L\left(FN_L\left(L_{att}\right) + L_{enc}\right).$$

The FN maps the input to a higher dimension d, enabling mutual engagement between the text and label.

This study adopts a case normalization method that differs from that reported in a previous study (Liu et al., 2022) in the processing of text features. This method not only avoids dependence on a small range of neurons but also maintains the independence of text instances and accelerates the convergence speed of the model regardless of the channel or batch size. The normalization formula of IN is expressed in equations as follows:

$$y_{tijk} = \frac{x_{tijk} - \mu_{ti}}{\sqrt{\sigma_{ti}^2 + \in}},$$

$$\mu_{ti} = \frac{1}{HW} \sum_{l=1}^{W} \sum_{m=1}^{H} x_{tilm},$$

$$\sigma_{ti}^2 = \frac{1}{HW} \sum_{l=1}^{W} \sum_{m=1}^{H} \left(x_{tilm} - mu_{ti}\right)^2.$$

For the selection of the activation functions, GELU (Lee, 2023) were selected. GELU is a high-performance neural network activation function that has been successfully applied to the BERT model (Hendricks & Gimpel, 2016). GELU functions exhibit excellent generalization and stable optimization abilities, which can improve model performance and reduce the difficulty and time cost of model training. The mathematical expression of the GELU is reproduced as follows:

$$GELUs(x) = xP(X \leq x) = x\Phi(x),$$

where $\Phi(x)$ denotes the probability function of a normal distribution.

To fully exploit the information encoded by the text engaged by the label and the relevance encoded by the label engaged by the text. In this study, a fusion encoder layer was introduced into the model and two mutually independent BiLSTM layers were constructed to propagate the fused text and label information. One BiLSTM layer was used to generate the ultimate text representation X_{fin} by combining the text encoding X_{fu} as follows:

$$X_{fin} = BLM_X\left(X_{fu}\right),$$

where $X_{fin} \in R^{m \times d}$. The subsequent decoding process is preserved in the hidden state $h \in R^{d \times 1}$ and cell state $c \in R^{d \times 1}$ of BLM_X. These are utilized to initialize the hidden and cell states of the LSTM decoder, which assists in generating a logical output from the input sequence.

Another BiLSTM encoder fuses the label encoding to produce a sequence of the label $L_{fin} \in R^{c \times d}$ as follows:

$$L_{fin} = BLM_L\left(L_{fu}\right).$$

ALD

For the final component of the model, we employed an ALD architecture. ALD consists of two steps for each time step. First, using the LSTM decoder, the hidden, cell, and loop context states were acquired in the first step of ALD decoding. Second, the probability of each class was determined using this component, which handles the classification problem simultaneously without modifying the model (*Yang et al., 2018*).

In the first step, we used LSTM cell attention as the benchmark technique for implementing the LSTM decoder. During training, we initially locate the label embeddings e_{t-1} of the $(t-1)$ decoding step in the true label and the predicted label embedding. We will use this alignment during the prediction process. Thereafter, the LSTM cell takes as input e_{t-1}, the recurrent context state r_{t-1}, the hidden layer state h_{t-1}, and the previous time step of the cell state c_{t-1}. It outputs the hidden state h_t and cell state c_t of the current time step t, as expressed in the following equation.

$$h_t, c_t = LSTMCell\left(\left[e_{t-1}; r_{t-1}\right], h_{t-1}, c_{t-1}\right),$$

where h and c denote the coding processes to initialize h_0 and c_0, respectively.

And then initialize e_0 and r_0. Once we obtained the hidden state h_t, we calculated the result of a_t when the step size is t. The adaptive classifier receives the hidden state h_t for subsequent processing.

$$a_t = Softmax\left(X_{fin} W_1 h_t\right),$$

Table 1 COVID dataset distribution.

Labels	0	1
Number of instances	35,700	64,300

where W_1 denotes a trainable matrix, and r_t is the state of the context at time step t, as expressed in the following equation.

$$r_t = \mathrm{Tanh}\left(W_2\left[X_{fin}^{T}a_t; h_t\right]\right),$$

where $W_2 \in R^{d\times 2d}$ represents a trainable matrix.

We used an adaptive classifier in the final classification phase to utilize the label representation of information-text participation. In comparison with most existing methods, the adaptive classifier can directly output the probability of each class by focusing on the label representation of text participation.

Label probability prediction

Given the hidden state h_t of the step size t, this classifier considers the final label of the code L_{fin} and the hidden state h_t as inputs to obtain the probability $\hat{y}_t$ of the time-step t.

$$\hat{y}_t = Softmax\left(L_{fin}W_3h_t\right),$$

where W_3 represents a trainable matrix and $\hat{y}_t$ is the output probability in each class. Therefore, to simplify the classification process, we integrated label representation into the text-attention mechanism.

After obtaining the probabilities for all the categories, we optimized our model by computing the objective function.

RESULTS AND DISCUSSION

Dataset

An experimental dataset called the COVID dataset was constructed by crawling COVID online reviews from December 2019 to March 2022 using Python Creeper technology. The constructed COVID dataset totaled 100,000 records. In the COVID dataset, Label 0 was designated as a negative sample, whereas Label 1 was considered a positive sample. The COVID dataset for the number of positive and negative sample distributions is shown in Table 1.

We also used Baidu's AI open platform (https://ai.baidu.com) to conduct a series of early stages. First, data capture. Large amounts of data are collected from various online web pages. By adjusting parameters such as keywords, page numbers, results per page and domain Settings, Baidu ensures that comprehensive data is collected according to specific needs. Second, handle paging. To crawl multiple pages of search results, the crawl script systematically increases page parameters to cover the required number of pages. Finally, preliminary data cleaning. This includes deleting duplicate data and detecting errors.

Table 2 **Dataset partial sample table.**

Label	Example
Positive	Now the sky, like a severe epidemic, envelops the city, and wishes the haze of the epidemic to withdraw as soon as possible.
Negative	What is wrong with people now, understanding that travel agencies want to make money, shouldn't we advocate safe travel at this time.

To better analyze the data, further data cleansing was done, such as removing Spaces, URL address information, and hashtags. Doing so will prepare you for the subsequent preprocessing phase.

The purpose of the text preprocessing stage is to carry out a series of processing and conversion of the cleaned text in order to facilitate the subsequent development feature extraction and model training. Data preprocessing is divided into two steps. The first step is text segmentation. Using Jieba word segmentation method deals with the review data, which is of great help to some new words and words not included in the dictionary. Step two, build stop word list. This article refers to the common stop word list after adding a custom stop word, in order to better handle comments on the data. Each preprocessed COVID dataset included the comments and the corresponding sentiment label. The data is shown in part for example, see Table 2.

Evaluation indicators

We employed the micro-averaged F1 score (Mi-F1) to evaluate the performance of our model. The Mi-F1 is a common metric used in evaluating classification models, especially in scenarios where imbalanced class distributions are present. The micro-averaging method treats the contributions of all categories equally, making it suitable for datasets with imbalanced sample distributions, as it calculates overall performance by aggregating the results across all categories. In practical applications, the Mi-F1 pays more attention to categories with a large number of instances, which is particularly important in applications such as text classification. The calculation method of the Mi-F1 is straightforward, easy to understand, and implement, making it a widely adopted tool among researchers and practitioners.

For category i, true positives are denoted as TP_i, false positives as FP_i, and false negatives as FN_i. The calculation formula is expressed as follows:

$$Mi-F1 = \frac{\sum_{i=1}^{c} 2TP_i}{\sum_{i=1}^{c} 2TP_i + FP_i + FN_i}.$$

Model parameter setting

In this section, we concatenated all the labels in a randomly selected order. The resulting label sequence was identical for all samples in the dataset and included every label in the label set. Using the same label sequence for all the samples, the model learned to

Table 3 **Model parameter setting.**

Experimental environment	Specific content
Word embedding	$d_{emb} = 400$
Internal model dimension	$d = 256$
BiLSTM output dimension	$d/2$
Hidden size of coordinate attention layer	$d_h = 100$
Attention headcount	3
Optimizer	Adam, the warmup schedule

recognize and classify each label consistently across the entire dataset. This can enhance the performance and provide more reliable results. The experimental settings used in this study are listed in Table 3.

Experimental result

Large pre-training models, such as BERT (*Devlin et al., 2018*), XLNet (*Yang et al., 2019*), ERNIE 3.0 (*Sun et al., 2021*), and NABoE (*Yamada & Shindo, 2019*), have recently gained popularity and have also acquired powerful language representation capabilities using large-scale unsupervised corpora. Some of these models outperformed our model; however, the number of parameters in these pre-trained models was large, making them less computationally efficient. For instance, the ERNIE 3.0 pre-training model exhibited the best performance but with billions of parameters. The BERT pre-training model had relatively few parameters; however, the number of parameters reached 340 million. Figure 2 shows the performance of the model in this study compared with four large pre-training models. We can clearly observe that the proposed model exhibits certain shortcomings in terms of performance. The pre-training model with the lowest number of parameters had approximately 68 times the number of parameters of the proposed model. In real-world scenarios, where time and space constraints are critical, the sheer number of parameters required by pre-training models can pose a significant disadvantage. These models often require extensive computational resources and are difficult to deploy in practical applications with limited resources. Therefore, it is important to balance model performance with practical constraints when selecting a mode for a given task.

We conducted an experimental evaluation of the proposed model in comparison with several baseline models on the COVID dataset, and compares the time complexity of each model, the performance of the prediction algorithm in large-scale data and resource use efficiency has a larger advantage. Table 4 summarizes the results of this comparison.

The DocBERT model (*Adhikari et al., 2020*) demonstrates superior performance in document classification by fine-tuning the BERT model.

The LSTM model (*Chen, Tseng & Wang, 2021*) possesses a sophisticated architecture that effectively manages long-term dependencies. Through the integrated function of its input, output, and forget gates, it dynamically controls the retention and omission of information within an LSTM unit at any given moment.

The Very Deep Convolutional Networks for Text Classification (VDCNN) (*Conneau et al., 2016*) employ minimal convolutional and pooling operations for character-based

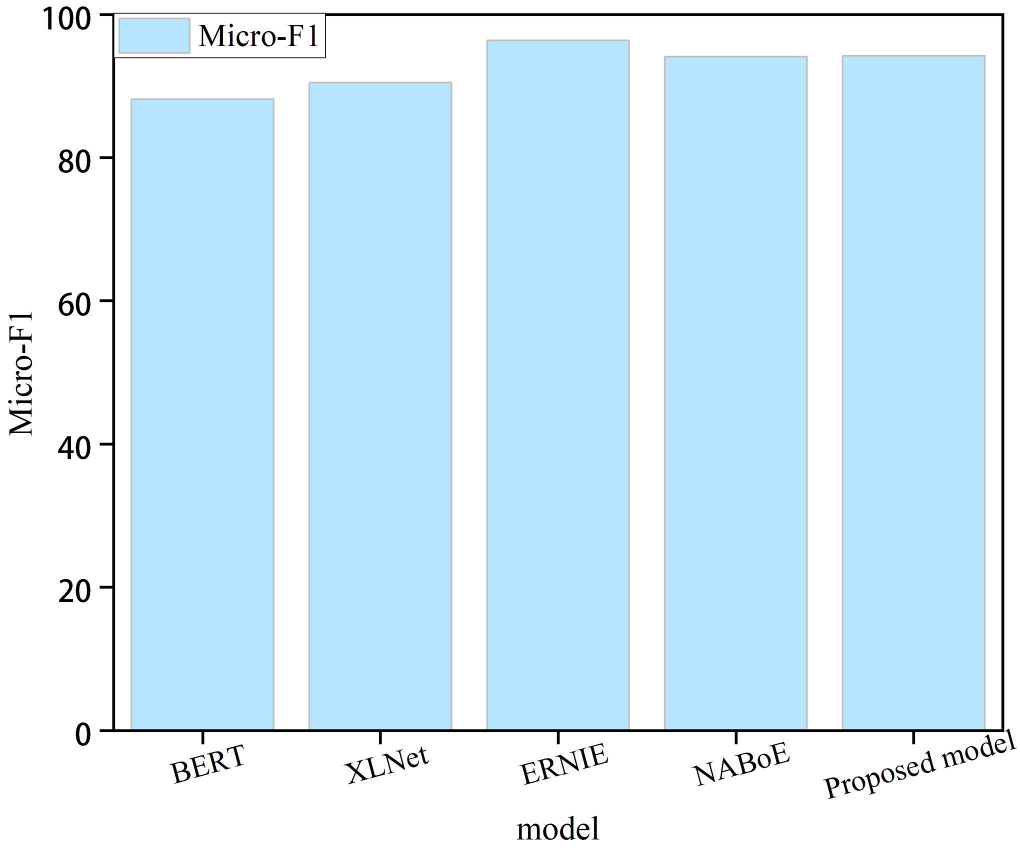

Figure 2 Comparison with the large pre-training model.
Full-size DOI: 10.7717/peerjcs.2240/fig-2

Table 4 Comparison results.

Model	Mi-F1 score	Times
DocBERT	89.40	4 h
LSTM	86.06	2.5 h
VDCNN	90.21	3.8 h
HTTN	92.10	2.3 h
LEAM	91.75	2 h
LSAN	90.00	2 h
CNLE	93.47	1.8 h
Proposed model	94.18	1.7h

classification, enabling the model to incorporate 29 convolutional layers, thus facilitating deep-text classification.

The HTTN model (*Xiao et al., 2021*) introduces a novel head-to-tail network that capitalizes on the relationship between the head and tail labels to facilitate the transfer of meta-knowledge from data-rich tail labels to those with less data.

The LEAM model (*Wang et al., 2018*) innovatively embeds each label within the same vector space as the word vectors, utilizing an attention mechanism to gauge the compatibility between the text sequence and the label embeddings.

The LSAN model (*Xiao et al., 2019*) represents a label-specific attention network optimized for multi-label text classification tasks. It effectively exploits the semantic relationships between labels and words to enhance classification accuracy.

The CNLE model (*Liu et al., 2022*) deeply integrates sequence information from both labels and texts. This integration captures comprehensive representations by amalgamating input from both the text and labels, thereby enhancing the precision and comprehensiveness of the classification.

As shown in Table 4, label embedding-based methods (HTTN, LEAM, and LSAN) generally surpass text representation methods (DocBERT, LSTM, VDCNN), underscoring the effectiveness of label-oriented strategies in improving model performance.

The CNLE model utilizes labeled representations in conjunction with textual interactions, allowing for the labeling of text engagements and thereby enriching the data. This improvement is achieved through the IN method, which independently normalizes each sample, reducing variance across different input data distributions. This normalization significantly enhances the model's ability to generalize across diverse datasets. Additionally, IN accelerates training convergence and improves stability by mitigating the Internal Covariate Shift (ICS)—the impact of input data distribution changes on model training. Therefore, the model described in this manuscript employs the IN method to enhance its generalization capabilities and stabilize the training process. Moreover, by integrating sequence information from both text and labels, IN helps the CNLE model process long textual data more effectively and capture contextual subtleties.

This study introduces the Gaussian Error Linear Unit (GELU) as the activation function in the proposed model to facilitate nonlinear improvements. The GELU function, by allowing the passage of small negative inputs, enables richer nonlinear transformations and the formation of more complex decision boundaries compared to the traditional ReLU. This is particularly beneficial in text categorization, where identifying subtle textual nuances is crucial. Within the CNLE framework, the GELU function enhances the interaction between text and label data, thus increasing the expressive power of the model's features. This leads to more accurate text classification, especially in cases involving ambiguous or complex classification criteria.

In summary, the integration of the CNLE-based IN method and the GELU function significantly boosts the effectiveness of the text categorization model discussed in this paper. This approach is particularly effective in managing diverse and extensive textual data, markedly improving both the accuracy of classifications and the robustness of the model.

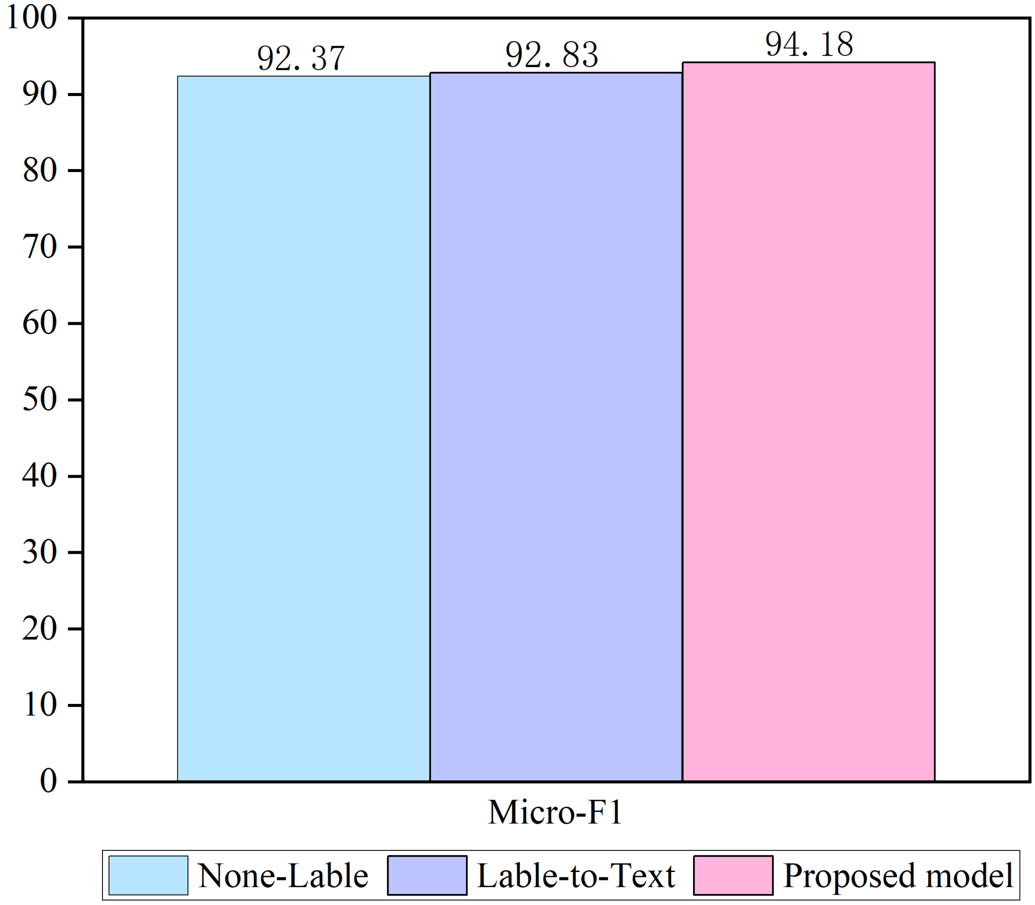

Figure 3 **Ablation of the model.**

Full-size DOI: 10.7717/peerjcs.2240/fig-3

Ablation studies

Two ablation studies are conducted to test the validity of the proposed model. The results are shown in Fig. 3.

A none-labeled model was used in the first ablation study, where only a sequence of texts was used as the input to the model. Most of the existing models similarly approach this method. In the second ablation study, both label embedding and label text attention components were retained in our approach (referred to as label-to-text). However, text label attention has been eliminated in our approach, ensuring that no text is incorporated in the label embedding. Our approach incorporates a text-label coordinate attention architecture, and the ablation results demonstrate that using a label-attention text representation and text-attention label representation can effectively enhance classification accuracy. This finding highlights the importance of considering both text and label components and suggests that incorporating a coordinated approach between the two can enhance performance.

Table 5 **Effect of the number of heads.**

Attention headcount	Mi-F1 score
1	91.02
2	93.54
3	94.18
6	92.12
8	92.91

Table 6 **Effect of the number of hidden sizes.**

Hidden size	Mi-F1 score
100	94.18
200	93.68
300	93.05

Parameter analysis

In this section, we experimentally explored the effects of the hyperparameters on the overall performance of the proposed model. The hyperparameters involved in the experiment primarily included the number of attention heads in the multi-head attention layer and the size of the hidden layer in the coordinate attention mechanism.

The number of attention heads in the multi-head attention mechanism is an important hyperparameter, which determines the number of positions on which the model can focus. Increasing the number of attention heads can improve the expression and learning ability of the model such that the model can learn the relationship between different positions and features simultaneously to better capture the information of the input sequence. However, too many attention heads can lead to overfitting or performance degradation; therefore, an appropriate selection is required. The experimental results are listed in Table 5.

The data presented in Table 5 clearly illustrates that the performance of the proposed model is suboptimal when the attention head count is set to one. This diminished efficacy is attributable to the reduction of the multi-head attention mechanism to a single original attention model, which compromises the precise allocation of weight information across different positions. As the number of attention heads increases, there is a corresponding improvement in model performance, reaching an optimal state when the count is three. However, an escalation in the number of attention heads beyond this point results in a deterioration of performance, indicative of model overfitting. This phenomenon suggests that a higher number of attention heads may lead to excessive model complexity, which negatively impacts generalization.

Thereafter, we evaluated the effect of the hidden size of the coordinate attention mechanism, ranging from 100 to 300, on the performance of the model. The experimental results are listed in Table 6.

As summarized in Table 6, the model performance gradually decreased as the size of the hidden layer in the coordinate attention mechanism increased. The decrease in

Table 7 The performance of the model under different dataset.

Dataset	Mi-F1 score
Yelp polarity	75.50
Amazon polarity	81.70

performance is attributed to the increase in the dimension of word representation, which causes an increase in the training difficulty of the model, resulting in underfitting.

Case study

In this section, experiments are conducted on two datasets, Yelp Polarity Reviews and Amazon Polarity Reviews, to further evaluate the model.

The Yelp Polarity Reviews dataset is extensively utilized in natural language processing and machine learning research, particularly for sentiment analysis. This dataset includes restaurant reviews from Yelp users, each annotated with either a positive or negative label, representing positive or negative sentiment respectively. It comprises over 500,000 reviews, evenly distributed between positive and negative sentiments.

Similarly, the Amazon Polarity Reviews dataset is frequently employed for sentiment analysis tasks. This dataset consists of product reviews from Amazon users, each labeled with positive or negative sentiment. It contains millions of reviews, providing a substantial resource for analyzing customer feedback and sentiment trends.

The experimental results of our proposed model for text classification on these two datasets are presented in Table 7.

Due to the lengthy training times required by deep learning models, the outcomes derived from various hyperparameters across different datasets will vary. Table 7 illustrates that the results achieved using the hyperparameters mentioned in the previous subsection are suboptimal for both datasets. To improve these results, extensive experimentation is necessary to further refine the model's hyperparameters. However, this paper does not include extensive hyperparameter tuning experiments for these datasets.

CONCLUSION

In this study, an optimized coordinate attention mechanism model was proposed to classify positive and negative samples to build a more efficient text classification model. First, we used label-attention text representation and text-attention label representation to obtain a shared representation of text and label sequences. By combining information from the text and labels, the model emphasized the relevant segments of both to a greater extent to perform text classification tasks better. Second, based on the self-attention model, IN and GELU functions were used, the convergence of the model was accelerated, and its performance was improved. Finally, using an adaptive decoder, we classified the comment text without modifying the model. Numerous experiments have indicated that the performance of the proposed method surpasses previous standard approaches.

This study was limited to the binary classification task of text and did not consider the needs of each scenario. Therefore, in the future, we will study the correspondence between labels in several task scenarios, apply them to multilabel classification tasks, and explore their advantages and disadvantages. It is also possible to extend the concept of coordinated attention mechanisms from text classification to other natural language processing tasks, such as natural language reasoning, dialogue systems, and language translation. By implementing such a mechanism, the performance of these NLP tasks can be improved by enabling the model to better capture the dependencies. We hope that the proposed method will promote research on text classification tasks in natural language processing and other fields, or consider adding discussion of potential limitations and future work to provide a balanced view of research contributions and areas of further investigation.

ADDITIONAL INFORMATION AND DECLARATIONS

Funding

This research is supported by the Shandong Social Science Planning Fund Program (Grant No. 21BTQJ02). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Grant Disclosures

The following grant information was disclosed by the authors:
The Shandong Social Science Planning Fund Program: 21BTQJ02.

Competing Interests

The authors declare there are no competing interests.

Author Contributions

- Rong Zhu conceived and designed the experiments, authored or reviewed drafts of the article, and approved the final draft.
- Hua-Hui Gao conceived and designed the experiments, performed the experiments, analyzed the data, performed the computation work, prepared figures and/or tables, and approved the final draft.
- Yong Wang performed the experiments, analyzed the data, performed the computation work, prepared figures and/or tables, and approved the final draft.

Data Availability

The following information was supplied regarding data availability:

Data and code are available in the Supplemental Files.

Supplemental Information

Supplemental information for this article can be found online at http://dx.doi.org/10.7717/peerj-cs.2240#supplemental-information.

REFERENCES

Adhikari A, Ram A, Tang R, Hamilton WL, Lin J. 2020. Exploring the limits of simple learners in knowledge distillation for document classification with DocBERT. In: *Proceedings of the 5th workshop on representation learning for NLP*. 72–77.

Alhutaish R, Omar N. 2015. Arabic text classification using k-nearest neighbour algorithm. *International Arab Journal of Information Technology* **12(2)**:190–195.

Chen C-W, Tseng S-P, Wang JF. 2021. Outpatient Text Classification System Using LSTM. *Journal of Information Science and Engineering* **37**:365–379 DOI 10.6688/jise.202103_37(2).0006.

Conneau A, Schwenk H, Barrault L, Lecun Y. 2016. Very deep convolutional networks for text classification. ArXiv arXiv:1606.01781.

Devlin J, Chang M-W, Lee K, Toutanova K. 2018. Bert: pre-training of deep bidirectional transformers for language understanding. ArXiv arXiv:1810.04805.

Du C, Chen Z, Feng F, Zhu L, Gan T, Nie L. 2019. Explicit interaction model towards text classification. *Proceedings of the AAAI conference on artificial intelligence* **33(01)**:6359–6366 DOI 10.1609/aaai.v33i01.33016359.

Graves A, Schmidhuber J. 2005. Framewise phoneme classification with bidirectional LSTM networks. *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005* **4**:2047–2052 DOI 10.1109/IJCNN.2005.1556215.

Gweon H, Schonlau M. 2024. Automated classification for open-ended questions with BERT. *Journal of Survey Statistics and Methodology* **12(2)**:493–504 DOI 10.1093/jssam/smad015.

Hendricks D, Gimpel K. 2016. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. ArXiv arXiv:1606.08415.

Ibrahim M, Gauch S, Salman O, Alqahtani M. 2021. An automated method to enrich consumer health vocabularies using GloVe word embeddings and an auxiliary lexical resource. *Peerj Computer Science* **7**:e668 DOI 10.7717/peerj-cs.668.

Kamyab M, Liu G, Adjeisah M. 2021. Attention-based CNN and Bi-LSTM model based on TF-IDF and GloVe word embedding for sentiment analysis. *Applied Sciences-Basel* **11(23)**:11255 DOI 10.3390/app112311255.

Lee M. 2023. Mathematical analysis and performance evaluation of the GELU activation function in deep learning. *Journal of Mathematics* **2023** DOI 10.1155/2023/4229924.

Li Q, Li P, Mao K, Lo EY-M. 2020. Improving convolutional neural network for text classification by recursive data pruning. *Neurocomputing* **414**:143–152 DOI 10.1016/j.neucom.2020.07.049.

Liu M, Liu L, Cao J, Du Q. 2022. Co-attention network with label embedding for text classification. *Neurocomputing* **471**:61–69 DOI 10.1016/j.neucom.2021.10.099.

Lu J, Batra D, Parikh D, Lee S. 2019. Vilbert: pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In: *Advances in Neural Information Processing Systems 32*.

Lu J, Yang J, Batra D, Parikh D. 2016. Hierarchical question-image co-attention for visual question answering. In: *Advances in Neural Information Processing Systems 29*.

Malik S, Jain S. 2024. Deep convolutional neural network for knowledge-infused text classification. *New Generation Computing* **42(1)**:157–176 DOI 10.1007/s00354-024-00245-6.

Qu H-Q, Kao C, Hakonarson H. 2024. Single-Cell RNA sequencing technology landscape in 2023. *Stem Cells* **42(1)**:1–12 DOI 10.1093/stmcls/sxad077.

Sun Y, Wang S, Feng S, Ding S, Pang C, Shang J, Liu J, Chen X, Zhao Y, Lu Y, Liu W, Wu Z, Gong W, Liang J, Shang Z, Peng S, Liu W, Ouyang X, Yu D, Tian H, Wu H, Wang H, Baidu Inc. 2021. Ernie 3.0: large-scale knowledge enhanced pre-training for language understanding and generation. ArXiv arXiv:2107.02137.

Sung YW, Park DS, Kim CG. 2023. A study of BERT-based classification performance of text-based health counseling data. *Cmes-Computer Modeling in Engineering & Sciences* **135(1)**:795–808 DOI 10.32604/cmes.2022.022465.

Ulyanov D, Vedaldi A, Lempitsky V. 2016. Instance normalization: the missing ingredient for fast stylization. ArXiv arXiv:1607.08022.

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I. 2017. Attention is all you need. In: *Advances in Neural Information Processing Systems 30*.

Wan CH, Lee LH, Rajkumar R, Isa D. 2012. A hybrid text classification approach with low dependency on parameter by integrating K-nearest neighbor and support vector machine. *Expert Systems with Applications* **39(15)**:11880–11888 DOI 10.1016/j.eswa.2012.02.068.

Wang G, Li C, Wang W, Zhang Y, Shen D, Zhang X, Henao R, Carin L. 2018. Joint embedding of words and labels for text classification. ArXiv arXiv:1805.04174.

Wu D, Wang Z, Zhao W. 2023. XLNet-CNN-GRU dual-channel aspect-level review text sentiment classification method. *Multimedia Tools and Applications* **83**:5871–5892 DOI 10.1007/s11042-023-15026-4.

Wu H, Zhou H, Zhou B, Wang M. 2023. SCMcluster: a high-precision cell clustering algorithm integrating marker gene set with single-cell RNA sequencing data. *Briefings in Functional Genomics* **22(4)**:329–340 DOI 10.1093/bfgp/elad004.

Xiao L, Huang X, Chen B, Jing L. 2019. Label-specific document representation for multi-label text classification. In: *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*. 466–475.

Xiao L, Zhang X, Jing L, Huang C, Song M. 2021. Does head label help for long-tailed multi-label text classification. *Proceedings of the AAAI Conference on Artificial Intelligence* **35(16)**:14103–14111 DOI 10.1609/aaai.v35i16.17660.

Xu S. 2018. Bayesian Naive Bayes classifiers to text classification. *Journal of Information Science* **44(1)**:48–59 DOI 10.1177/0165551516677946.

Yamada I, Shindo H. 2019. Neural attentive bag-of-entities model for text classification. ArXiv arXiv:1909.01259.

Yang P, Sun X, Li W, Ma S, Wu W, Wang H. 2018. SGM: sequence generation model for multi-label classification. ArXiv arXiv:1806.04822.

Yang Z, Dai Z, Yang Y, Carbonell J, Salakhutdinov RR, Le QV. 2019. Xlnet: generalized autoregressive pretraining for language understanding. In: *Advances in Neural Information Processing Systems 32*.

Yu Z, Yu J, Fan J, Tao D. 2017. Multi-modal factorized bilinear pooling with co-attention learning for visual question answering. In: *Proceedings of the IEEE international conference on computer vision*. Piscataway: IEEE, 1821–1830.

Zainuddin N, Selamat A, Ibrahim R. 2018. Hybrid sentiment classification on twitter aspect-based sentiment analysis. *Applied Intelligence* **48(5)**:1218–1232 DOI 10.1007/s10489-017-1098-6.

Zhang P, Wu H. 2023. IChrom-deep: an attention-based deep learning model for identifying chromatin interactions. *IEEE Journal of Biomedical and Health Informatics* **27**:4559–4568 DOI 10.1109/jbhi.2023.3292299.

Zhu Y. 2021. Research on news text classification based on deep learning convolutional neural network. *Wireless Communications & Mobile Computing* **2021**:1508150 DOI 10.1155/2021/1508150.