Cognitive Computation (2024) 16:740–755
https://doi.org/10.1007/s12559-023-10237-1

A Cognitively Inspired Multi-granularity Model Incorporating Label Information for Complex Long Text Classification

Li Gao[1] · Yi Liu[2] · Jianmin Zhu[3] · Zhen Yu[4]

Received: 13 April 2023 / Accepted: 16 December 2023 / Published online: 26 December 2023

Abstract

Because the abstracts contain complex information and the labels of abstracts do not contain information about categories, it is difficult for cognitive models to extract comprehensive features to match the corresponding labels. In this paper, a cognitively inspired multi-granularity model incorporating label information (LIMG) is proposed to solve these problems. Firstly, we use information of abstracts to give labels the actual semantics. It can improve the semantic representation of word embeddings. Secondly, the model uses the dual channel pooling convolutional neural network (DCP-CNN) and the timescale shrink gated recurrent units (TSGRU) to extract multi-granularity information of abstracts. One of the channels in DCP-CNN highlights the key content and the other is used for TSGRU to extract context-related features of abstracts. Finally, TSGRU adds a timescale to retain the long-term dependence by recuring the past information and a soft thresholding algorithm to realize the noise reduction. Experiments were carried out on four benchmark datasets: Arxiv Academic Paper Dataset (AAPD), Web of Science (WOS), Amazon Review and Yahoo! Answers. As compared to the baseline models, the accuracy is improved by up to 3.36%. On AAPD (54,840 abstracts) and WOS (46,985 abstracts) datasets, the micro-F1 score reached 75.62% and 81.68%, respectively. The results show that acquiring label semantics from abstracts can enhance text representations and multi-granularity feature extraction can inspire the cognitive system's *understanding* of the complex information in abstracts.

Keywords Text classification · Neural network · Machine learning · Multi-head attention · Gated recurrent unit

Introduction

Cognitive systems help humans *understand* textual information from the outside world and acquire the corresponding knowledge. Artificial simulation of this cognitive process is beneficial to explain such cognitive phenomena [1]. Natural language processing (NLP) uses computers to *understand* human language, bringing machines closer to human cognitive systems. Text classification is one of the cognitively inspired methods in NLP. Text classification technology based on neural networks simulates human brain structure and cognitive processing [2], giving computers the ability to perform corresponding cognitive tasks. It realizes automatic abstract classification by conducting big data analysis of resources within the discipline to *understand* abstracts comprehensively and extensively [3]. However, unlike the general text structure, abstracts involve a variety of natural sciences. As a result, complex labels and the lack of label information make it difficult to accurately map the text features to the corresponding labels space [4]. Not only that, abstracts have

✉ Li Gao
gaoli@usst.edu.cn

Yi Liu
222260564@st.usst.edu.cn

Jianmin Zhu
jmzhu@usst.edu.cn

Zhen Yu
Dt_yz2021@163.com

[1] Library & The Department of Computer Science and Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China

[2] The Department of Computer Science and Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China

[3] School of Mechanical Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China

[4] Shanghai Datong High School 200011, Shanghai, China

a higher degree of professionalism, the general word vector is difficult to express comprehensive semantic information [5]. Meanwhile, many supplementary explanations introduce lots of noise, which are not related to the topic. This noise makes the length of the abstracts longer and the features scattered. Therefore, higher requirements are put forward for cognitive systems to *understand* the relevance of context.

Most of the word embeddings used by existing methods for text classification are based on language models. For example, bidirectional encoder representations from transformers (BERT) uses unsupervised objectives and trained on large numbers of text data. Unlike other models, it uses bidirectional coding structures to enhance the generalization ability of pre-trained encoder models, which made great contributions in text classification. Recently, Moraangthem and Lee [6] considered a lite BERT (ALBERT) as a better pre-trained model using parameter reduction technology, which significantly reduces the amounts of parameters and improves performance of BERT. The high professionalism of abstracts can easily lead to the label semantics being far from the sample semantics because of the lack of relevant knowledge [7]. To solve the issues, this paper proposes a fusion label information model to generate label semantics by integrating sample information. On this basis, label semantics and text information are taken as two kinds of attention heads and multi-head self-attention realizes the feature interaction between labels and texts. It not only highlights the weight of professional features but also enhances the semantic representation ability of embeddings.

Apart from making the most of label information, methods for abstract classification are also critical. Due to the structural characteristics of abstracts in academic articles, there are many supplementary explanations. These explanations not only introduce excessive noise to interfere with the model mining text information, but also increase the length of the abstracts, causing long-term dependence problems [8]. Due to the length of abstracts, the local features extracted by traditional convolutional neural networks (CNN) are not comprehensive enough and the global semantic information contained in the long texts cannot be used [9]. While recurrent neural networks can extract global feature information, the high proportion of noise content in the texts causes a fragmented distribution of features. It is easy to affect the extracted global features. Nowadays, the traditional feature extraction methods can no longer adapt well to the classification task of text in the professional field, and it is urgent to design a highly professional text classification model for the abstracts. In this work, we design a dual channel pooling mechanism to improve CNN. The deep semantic information channel uses the maximum pooling to retain the maximum features of the sentences. It highlights the key content in the abstracts and avoids the key information being overwritten when the text length is too long. The average pooling method in the shallow semantic information channel retains the overall information of the sentence, which is suitable for the underlayer TSGRU to extract context-related features. TSGRU adds a timescale to recur the past features after filtering, which strengthens the long-term dependence between texts and improves the model's mining ability for potential features of texts. AAPD dataset contains 55,840 abstracts and each abstract contains about 200 to 500 words. WOS dataset collects abstracts from 46,985 articles published on the Web of Science. The two datasets are suitable for evaluating the performance of the model on abstracts with long length. Amazon Review and Yahoo! Answers datasets have the max length of 32,788 characters and 4000 characters, respectively. Therefore, they are suitable for evaluating the classification performance of longer texts.

The main contributions of this paper are as follows:

- In terms of pre-trained encoder model, we propose a method of fusing label information to improve the ability of abstracts representation. It uses multi-attention mechanism to integrate the sample public information as labels semantics and multi-head attention to combine labels and texts information.
- In terms of text classification model, we propose a multi-granularity model to solve the problem of excessive noise in abstracts and dispersive features. It introduces DCP-CNN to enhance the feature recognition of key features and the coverage of sequence information of the entire abstracts.
- Considering that CNN cannot effectively extract the spatial information of the abstracts, TSGRU is proposed to obtain more comprehensive spatial semantic information through the timescale and enhance the ability to suppress noise and retain the contextual semantic features through a soft thresholding mechanism.

This paper is organized as follows. The "Related Work" section presents the review of literature. The "Research Methodology" section presents the details of the proposed model. The "Experiments and Analysis" section shows the analysis and results of experiments. The "Discussion" section discusses the results of experiments and the "Conclusion" section summarizes the paper.

Related Work

Deep neural network models have obtained large success in many natural language processing tasks. These cognitively inspired models achieve satisfactory results in text classification with the optimization in different aspects and promote the development of the cognitive systems.

Table 1 Comparison of BERT and ALBERT

Model		Parameters	Layers	Embedding
BERT	Base	108M	12	768
	Large	334M	24	1024
	xlarge	1270M	24	2048
ALBERT	Base	12M	12	128
	Large	18M	24	128
	xlarge	59M	24	128

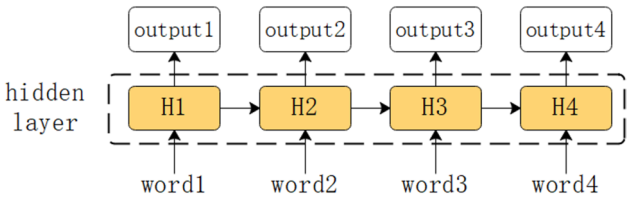


Fig. 1 Structure of RNN

Word Embedding

Language models using pre-trained word embedding matrices have higher training speed and accuracy than random word embedding matrices [10]. Glove's method of word representation based on count-based and overall statistics [11] reduces the amount of computation and storage space of data. BERT is a new language model [12] that targets the masked language model to predict the next sentence with masked or replaced words to generate deep bidirectional language representations. ALBERT reduces the amounts of parameters while maintaining performance and improving efficiency of parameters [13], the specific number of parameters is shown in Table 1.

Compared with BERT, ALBERT has a smaller number of parameters under the same conditions, and the classification performance is the same as BERT. The language model pre-trained by ALBERT can not only *understand* the semantics of texts accurately and break through the polysemy problem that static word vectors cannot solve, but also improve the operation efficiency of the model.

Text Classification

The traditional text classification method is to make multiple categories of features artificially, such as vocabulary, syntax and term frequency. Then put them into machine learning models, such as support vector machines (SVM), naive Bayes and random forest [14]. However, extracting features manually is a task that requires a lot of expertise, which omits long-term relationships in the text corpus and makes it difficult to cope with the fast-growing field of academic articles. CNN and Recurrent Neural Network (RNN) have long been popular. RNN is a class of neural networks to process sequence data. As shown in Fig. 1.

RNN treats the text as a sequence of words and *understand* the structure in it. However, in the face of long tests, the gradient vanishing will appear when the depth of the neural network is too deep. The practice and theory of gated units have long been studied. Long short-term memory (LSTM) first applied them to the hidden layers of RNN, controlling the flow of information through a gating mechanism to mitigate gradient vanishing. It is excellent at processing sequence data and easy to capture long-term and short-term dependencies [15]. However, this model cannot achieve key information in the text and it is hard to capture local features in the text. Gate recurrent unit (GRU) is similar to LSTM. It reduces the number of gating units under the premise of ensuring classification accuracy. Therefore, it is easier to train and improve training efficiency greatly. MTGRU builds on the GRU by increasing the above share through timescale, which strengthens the relevance of context. The gating units of LSTM, GRU and MTGRU are shown in Fig. 2.

The variants of LSTM and GRU can obtain overall semantic information [16]. Sentiment analysis uses interactive LSTM [17] to model interactions between individuals to discover changes in each person's emotional state. Bi-directional long short-term memory (Bi-LSTM) is used to obtain the global representation of the article. Combine with a multi-convolutional neural network (MCNN) to capture shallow features flexibly and use the attention mechanism to capture more comprehensive key information [18]. The use of LSTM with an attention layer [19] allows the network to select the most relevant feature for each label. A long text classification algorithm integrating multi-feature-level attention mechanism [20] is proposed, which uses bidirectional gated recurrent unit (Bi-GRU) and CNN to extract multiple feature fusions to obtain specific target vectors. Bi-GRU with attention mechanism and capsule network performs better when processing tasks with less data. At the same time, the correlation between words is preserved [21]. The effectiveness of timescale [22, 23] on neural networks has been demonstrated. On this basis, Moirangthem and Lee [6] proposed that hierarchical MTGRU to capture multiple compositions and enhanced the network's ability to model longer text sequences. In addition, Pal et al. [24] designed two new decoding units in the GRU to speed up convergence and added a new gating unit to reserve longer memory. Aote et al. [25] used the particle swarm optimization algorithm to process multiple features of the abstract and achieved good performance.

CNN has great advantages in parallel computing and has the ability to get local correlations and extract higher levels of correlation through pooling [26], which allows it to extract sentences from a continuous context window. Kaur [27] used CNN to improve its performance based on

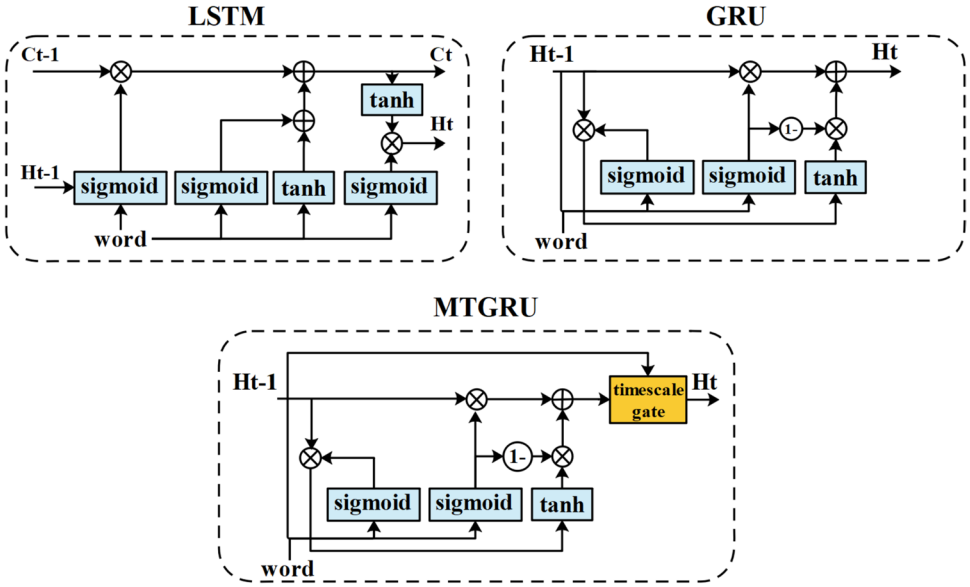


Fig. 2 Gate recurrent units of LSTM, GRU and MTGRU

BERT. Rafiepour et al. [28] used several convolutional layers with different kernel sizes to preserve the correspondence between tokens and labels. Liang et al. [29] utilized a combination of a well-designed multi-view representation learning and data transfer methods to extract and weight text with multi-granularity representations automatically. Ayetiran [30] used convolution operations to extract attention signals and highlight emotional words and flip words that focus on the text. Using char embedding as input to CNN avoids that traditional word embedding does not have a good effect on low-frequency words [31]. In addition, Li et al. [32] introduced inductive learning methods on the basis of graph convolution to enhance the interpretability of text information. The introduction of exogenous knowledge to build a network solved the problem that existing methods ignore the semantic and structural information of nodes effectively. However, using word frequency to measure the importance of words could not reflect sequence information and was easily affected by dataset skew.

Study on Labels

In addition to utilizing text representation, label information can be leveraged to improve text classification. The division of label hierarchy combines text-to-label attention and text labels participate in the representation [33]. Label information leverages the feedback of text representations to encode labels with more information. Wang et al. [34] established the interaction function between labels and texts through a multilayer perceptron and experiments proved that the information representation of labels can be effectively enhanced. But datasets often use fixed label annotations, ignoring relationships between labels. Qian et al. [35] proposed the label-level contrastive learning (LLCL) paradigm to constrain unreasonable label distribution and capture label correlation. Wang et al. [36] designed a guide network label strengthening strategy, which used label semantics to fine-tune the pre-trained classification model. But the model was only valid for labels with fixed semantics.

Research Methodology

This section describes the classification model in detail. The frame of the model is shown in Fig. 3.

Firstly, the label-text fusion obtains label semantics and enhances representation of abstracts with the use of label information. Secondly, the features of different granularities are extracted by DCP-CNN and TSGRU, and then the fusion gate realizes the information fusion of the two. Finally, the classifier outputs the prediction results according to the fusion features. The following sections will introduce the structure of the model in turn.

Pre-trained Encoder Layer

In this section, the fusion label information model is described in detail. Its purpose is to integrate label information into the encoding of text sequences, so that labels are more closely related to abstracts. As shown in Figs. 4 and 5, it is mainly composed of multi-attention semantic extraction and multi-head attention layer.

The multi-attention semantic extraction layer uses ALBERT to obtain word embeddings, and then puts samples into set S_i according to their corresponding labels, where i represents a label, $i \in [1, I]$, and I represents the number of labels. We use Eq. (1) to calculate the semantic similarity weight matrix between samples.

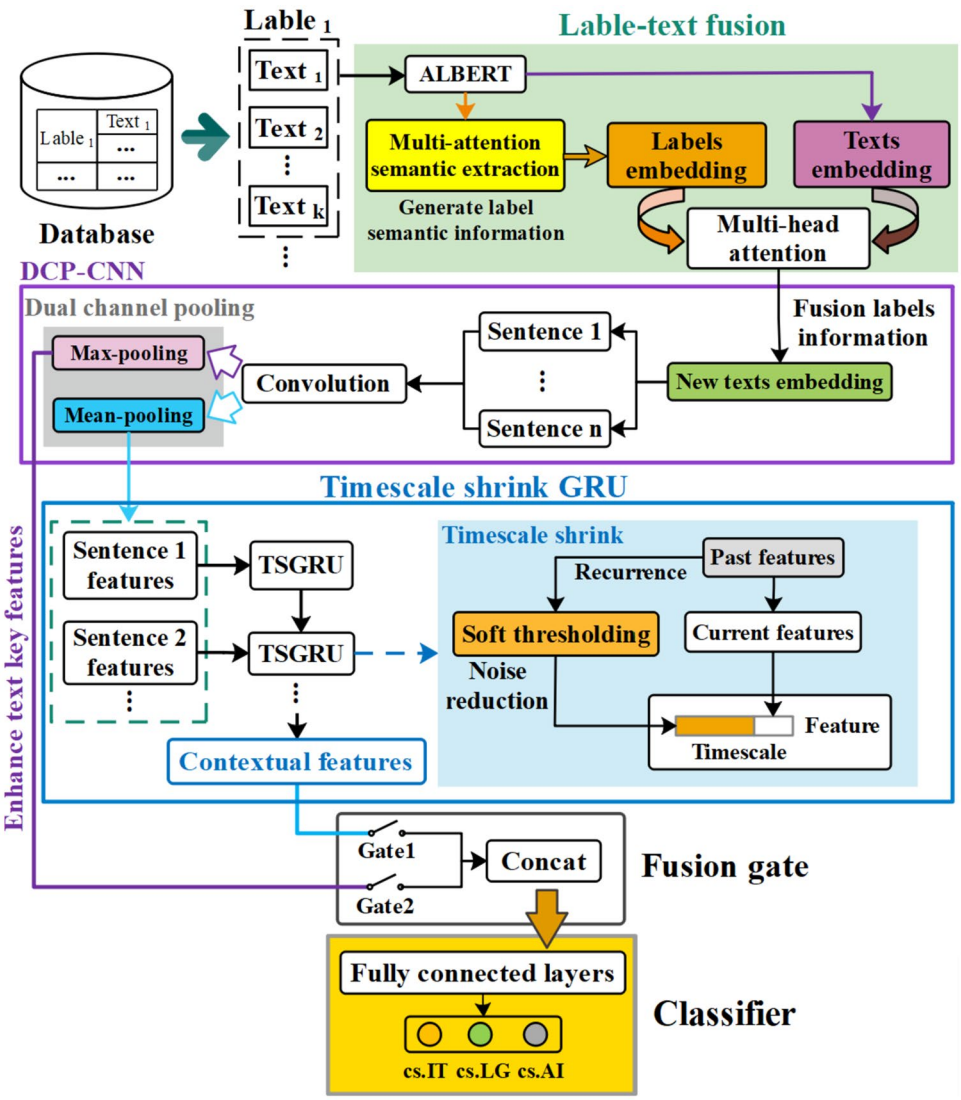


Fig. 3 The frame of the cognitively-inspired multi-granularity model incorporating label information (LIMG). Label-text fusion is used to improve representations of abstracts. DCP-CNN and TSGRU extract the features of different granularities. Fusion gate fuses multi-granularity features and puts them into classifier

$$\delta_x = \frac{Relu\left(d_x^{\lambda}\right)}{\sum_{n=1}^{L} Relu\left(d_n^{\xi}\right)} \tag{1}$$

where d_x^{λ} is a word embedding x from the sample λ, d_n^{ξ} is a word embedding from the sample ξ. Samples λ and ξ are different samples from the set S_i. Then L is the length of the sample and *Relu* represents the activation function. We use the *Relu* activation function because it has stronger nonlinear fitting ability and high computational efficiency. The computed weight δ_x of x helps the attention mechanism extract common semantics.

As shown in Fig. 4, the process is as follows: divide the samples from S_i into groups of two and use the attention mechanism to pay attention to their corresponding first-level intermediate semantics. Therefore, the non-common semantics between the two samples are weakened and the common semantics are retained. Then repeat the above steps to obtain more advanced intermediate semantics in all first-level intermediate semantic groups. Finally, the label semantics pointed to by this dataset are obtained.

The word embeddings of texts and labels obtained through the multi-attention semantic extraction layer can be expressed as $x_{emb} = \{x_1, x_2, x_3, \ldots, x_n\}$ and $l_{emb} = \{l_1, l_2, l_3, \ldots, l_c\}$, where $x_i \in R^{n*d}, l_i \in R^{c*d}$, n is the number of words in the texts, and c is the number of labels.

To obtain a textual representation containing label information, the multi-head attention layer helps the model pay more attention to label-related words. The scaled dot product attention is as follows [37]:

$$Attention(Q, K, V) = Softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{2}$$

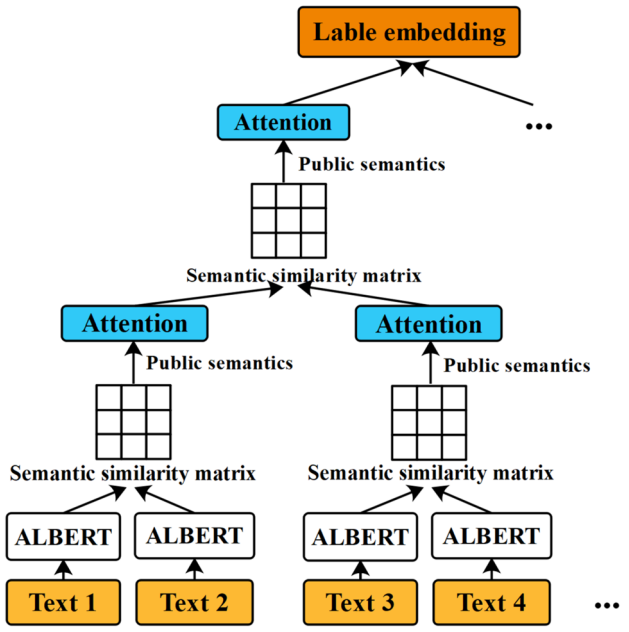


Fig. 4 Multi-attention semantic extraction

where $Q \in \mathbb{R}^{q \times d_k}$, $K \in \mathbb{R}^{k \times d_k}$, $V \in \mathbb{R}^{k \times d_v}$ and we set $d_k = d_v$. The definition of multi-head attention is as follows:

$$\begin{aligned} &MultiHead(Q,K,V) = Concat(H_1;\ldots;H_p)W^o \\ &where\ H_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \end{aligned} \tag{3}$$

where $W^o \in \mathbb{R}^{hd_h \times d_k}$, W_i^Q, W_i^K and $W_i^V \in \mathbb{R}^{d_k \times d_h}$. h is the number of heads and $i \in [1, h]$. The dimension of each head is $d_h = d_k / h$. *Concat* is used to connect the heads of multi-head attention. To make the model pay more attention to the words related to the label, we feed x_{emb} and l_{emb} into the multi-head attention module at the same time to get label-attended text representation X_{att} [33].

$$X_{att} = MultiHead(x_{emb}, l_{emb}, l_{emb}) \tag{4}$$

We use word embeddings as query vectors to calculate the relevance of word embeddings and labels. The word embeddings associated with labels obtain greater attention weight.

Sample: Select an abstract and the corresponding labels from the AAPD dataset. The text is from the abstract while stat.ME and cs.IR are the labels corresponding to this abstract. The stat.ME stands for methodology subject in the domain of statistics and cs.IR stands for information retrieval subject in the domain of computer science. Then we get the corresponding word embeddings x_{emb} and l_{emb} through ALBERT coding layer. The dimension of the word embeddings is 128 and the number of heads is 2. With the help of multi-attention semantic extraction layer, we can get the weights of the words related to the label. Figure 6 is the display after the visualization of the attention weights. The darker the color, the more relevant it is to the label cs.IR.

Finally, we use two independent Feed Forward Networks (FFN) and residual connections to get their fused encoding. After Layer Normalization (LN), we get fusion encoding X_{fuse}:

$$X_{fuse} = LN_X\left(FFN_X\left(X_{att}\right) + X_{emb}\right) \tag{5}$$

The obtained fusion encoding X_{fuse} will be classified in the multi-granularity classification model as word embeddings of the texts.

LIMG

Dual Channel Pooling CNN

Due to the high proportion of noise content in the abstracts, it is difficult to extract long text features while eliminating

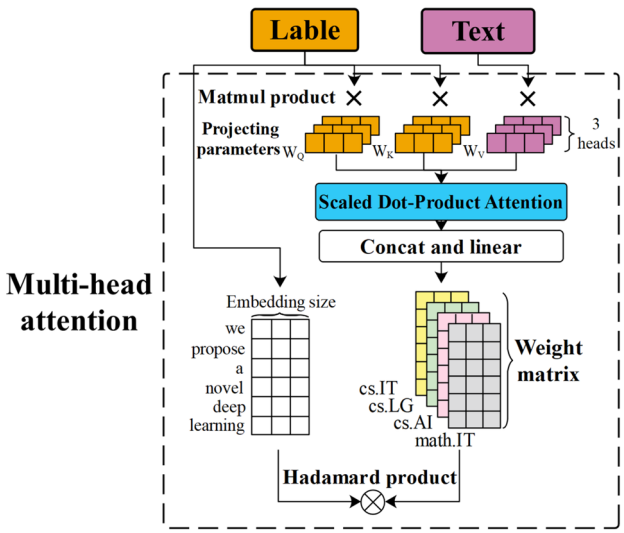


Fig. 5 Multi-head attention

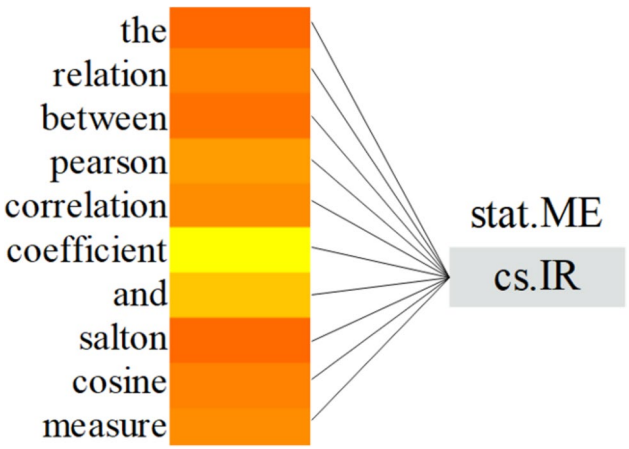


Fig. 6 Label-attended text encoding

the influence of noise only through the general shallow convolutional structure. Therefore, the abstracts are divided into sentences according to the hierarchical structure of the text and a dual channel pooling CNN is designed to extract local key information and context sequence information in the text. It can extract both local key information and contextual sequence information at the same time. Firstly, the sentence feature vectors c are extracted by CNN, and then the mean-pooling and max-pooling are performed in two channel dimensions respectively. We can obtain the feature vector c_{avg} containing the shallow semantic information of the texts and the feature vector c_{max} containing the deep semantic information, c^i_{avg} and c^i_{max} is as follows:

$$c^i_{avg} = \frac{1}{r}\sum_{k=1}^{r} c^i_k \tag{6}$$

$$c^i_{max} = max(c^i_1, c^i_2 \dots c^i_k) \tag{7}$$

where c^i_k denotes vector k in the sentence i when use the kernel size of m. c^i_{avg} denotes the vector after using mean-pooling in c^i_k and r is the number of these vectors. Then we connect c^i_{avg} of all sentences to get c_{avg}. Similarly, we replace mean-pooling with max-pooling to get c_{max}.

The shallow semantic information focuses on the general content of the abstracts and then it will be extracted by TSGRU. Deep semantic information focuses on the key content to compensate for the key information forgotten during the TSGRU extraction process. CNN can also flexibly set multiple convolution filters to extract deep semantic features. Features of different sizes are extracted separately by sliding on the X_{fuse} with different kernel sizes k (e.g., kernel = 1, kernel = 3, kernel = 5 in Fig. 7). Then, the max-pooling operation is performed to reduce the dimensionality of the features and extract more important information. On this basis, multi-head attention provides multiple subspaces to refine the distribution of attention weights. Each attention head can focus on measuring the weight of the word in the current position.

Algorithm Design of DCP-CNN

According to the above description of DCP-CNN, the following algorithm is designed. It uses X_{fuse} as input that is partially integrated into the label information in the "Data" section. Each sentence extracts semantic information through two channels: mean-pooling uses kernels of size 1 to retain complete sequence information and max-pooling uses different sizes of kernels to extract deep semantic information. The detailed procedure is shown in Algorithm 1.

The shallow semantic information h_{mp} is extracted by the underlayer TSGRU to make up for the missing sequence information of DCP-CNN.

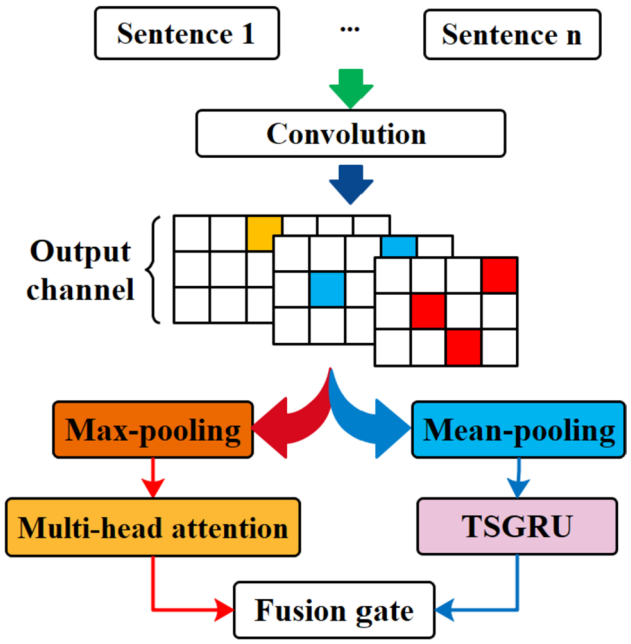


Fig. 7 Dual channel pooling CNN

Timescale Shrink Gated Recurrent Units

GRU solves the problem that CNN cannot extract temporal features and alleviates gradient vanishing, but with the increase in the length of abstracts, more and more past information disappears because of the gating units. This will destroy the long-term dependence in long texts, so adding a variable called timescale in GRU will increase the proportion of the past information and strengthen the context connection, to obtain more comprehensive global features. In order to enhance the resistance of the model to noise in texts, a soft thresholding algorithm is introduced to the timescale. Soft thresholding is a common algorithm in signal noise reduction processing. When the features are lower than the threshold, it can be considered that this part of the features is useless and will be zeroed out. The other part of the features will be retained. In this way, the noise reduction treatment can be achieved. The formula is as follows:

$$y\begin{cases} x-\lambda, x>\lambda \\ 0, -\lambda <= x <= \lambda \\ x+\lambda, x<-\lambda \end{cases} \tag{8}$$

x, y are input and output vectors, respectively. λ is the threshold.

The timescale is to add another constant gating unit to blend the features of current and past hidden states essentially.

Algorithm 1 Dual channels pooling CNN

Input: Word vectors x_n, number of filters filternum

Output: Hidden state h_{mp}, h_{max}

for epoch: = current epoch to max epoch

{

x_1 = Convolution (x_n, filternum, kernelsize = 1);

output_1 = Maxpooling (x_1);

output_mp = Meanpooling(x_1)

h_1 = Relu (output_1);

h_mp = Relu (output_mp);

x_2 = Convolution (x_n, filternum, kernelsize = 3);

output_2 = Maxpooling (x_2);

h_2 = Relu (output_2);

x_3 = Convolution (x_n, filternum, kernelsize = 5);

output_3 = Maxpooling (x_3);

h_3 = Relu (output_3);

output_4 = Concatanddropout (h_1, h_2, h_3);

h_{mp} = output_mp

h_{max} = Multheadattention(output_4);

return h_{mp}, h_{max};

}

Each step of the TSGRU takes x_t from h_{mp} and the previous hidden states h_{t-1} as input to obtain the output vectors h_t of the hidden layer. It contains a reset gate r_t and an update gate z_t to determine how many features of the past hidden state are retained [6], as is shown in Eq. (8):

$$r_t = \sigma\left(W_{xr}x_t + W_{hr}h_{t-1}\right)$$

$$z_t = \sigma\left(W_{xz}x_t + W_{hz}h_{t-1}\right)$$

$$u_t = tanh(W_{xu}x_t + W_{hu}(r_t \odot h_{t-1)})$$

$$\widetilde{h}_t = z_t h_{t-1} + \left(1 - z_t\right)u_t \tag{9}$$

where u_t and $\widetilde{h}_t$ serve as candidate activation and hidden state vectors of the current gating unit. σ(·) and tanh(·) are the sigmoid and tanh activation functions. ⊙ denotes the Hadamard product.

The timescale gating unit is shown in Eq. (10):

$$h_t = \widetilde{h}_t \frac{1}{\tau} + \left(1 - \frac{1}{\tau}\right)h_{t-1} \tag{10}$$

The constant τ is used to control the timescales of each TSGRU cells. On the one hand, larger τ increases the features of the previous text sequence, which makes the gated unit retain more long-term dependency. It is conducive to extracting features from longer texts. On the other hand, a smaller τ makes the scale factor 1/τ larger, so the current time series $\widetilde{h}_t$ accounts for more weight. The gating unit will contain more features of the current time series. The τ, like other weight parameters in neural networks, is a trainable variable that is optimized with the final loss.

Algorithm Design of TSGRU

Based on the above description, the algorithm of TSGRU to extract the global features of abstracts is shown in Algorithm 2.

The algorithm is called at every step of the training process. All parameters are initialized before training, including

hidden state vector h_{t-1} and the time scale parameter τ. We feed word embedding vectors x_t from the dual channel pooling CNN into the model. Then we can obtain h_{st} after filtering the noise in h_{t-1} by the soft thresholding algorithm. Filter h_{st} and x_t through the reset gate and update gate to get the candidate activation u_t. The τ is used to adjust u_t and h_{st} to get the next hidden state h_t. The update of the timescale τ starts after a specific number of batches of training. The final output of the algorithm h_t will be fused with the deep semantic information h_{max} in the next section.

Fusion Gate

Since the deep semantic information h_{max} enhanced by DCP-CNN and the global features h_t extracted by TSGRU may be complementary and duplicated, we used a gating unit to fuse the features from two aspects:

$$g_t = \sigma\left(W_g o_g + W_c o_c + b\right) \tag{11}$$

$$o_t = g_t o_g + \left(1 - g_t\right) o_c \tag{12}$$

g_t is the gating unit for selecting the features, o_g is the global features extracted by TSGRU, o_c is the deep semantic features extracted by the CNN and o_t is the text features filtered by the gating unit. Finally, the fully connected layer and activation of the classifier will output the probabilities of labels to which the abstracts belong.

Experiments and Analysis

Datasets

In order to comprehensively compare the performance between LIMG and the traditional classification models, four benchmark datasets are used to cover different text lengths and multiple classification tasks. The statistics summary of these datasets is shown in Table 2.

Arxiv Academic Paper Dataset (AAPD): Contain abstracts of 55,840 academic articles from the site. Each abstract involves multiple disciplines and the total number of disciplines is 54. Each abstract has multiple labels and each label has many samples. Each abstract contains about 200 to 500 words, which is suitable for evaluating our model.

Algorithm 2 Timescales shrink gated recurrent units

Input: Hidden state vectors h_{t-1}, word vectors x_n and current timescale τ

Output: Next hidden state h_t and updated timescale τ

if current epoch < max epoch then

  read growth factor;

  h_{st} = SoftThresholding(h_{t-1});

  rest gate r_t = GetRestgate(x_t, h_{st});

  update gate z_t = GetUpgate(x_t, h_{st});

  candidate activation u_t = GetCandidate(x_t, h_{st}, r_t);

  if the perplexity has not decreased for last 3 steps

  then

    τ = τ * growth factor;

    return τ ;

  else

    return τ ;

  end

  next hidden state $h_t = (z_t h_{st} + (1 - z_t)u_t)$ * 1/τ + h_{st} * (1-1/τ);

  return h_t;

end

Table 2 The details of the text classification datasets

Dataset	Classes	Training set	Testing set
AAPD	54	53,840	1000
WOS-46985	7	37,588	9397
Amazon F	5	3,000,000	650,000
Amazon P	2	3,600,000	400,000
Yahoo! Answers	10	1,400,000	60,000

WOS-46985: The Web of Science (WOS) dataset collects data such as abstracts, domains, and keywords from 46,985 articles published on the Web of Science. The categories of first-level include 7 categories of computer science, psychology, mechanical engineering, electrical engineering, biochemistry, medical science and civil engineering.

Amazon Review: Come from the Stanford Network Analysis Project (SNAP). The full dataset (Amazon F) includes 34,686,770 reviews on 2,441,053 products and the max length of reviews is 32,788 characters. Reviews are divided into 1–5 star representing user satisfaction. Amazon Review Polarity Dataset (Amazon P) is a subset that contains 3,600,000 training samples and 400,000 testing samples in 2 polarity sentiment.

Yahoo! Answers: Topic Classification from "Yahoo!" Corpus of answers. It contains the questions in the corpus and the related answers to them and the text length can be up to 4000 characters. It includes 10 classes, each containing 140,000 training samples and 5000 test samples, respectively.

Experiment Settings

Word embeddings with label information are used as input in the experiments. Among them, 128 units of TSGRU and 128 units of DCP-CNN are used to extract features. Following the parameter setting in Yun et al. (2022), the timescale parameter τ is initialized to the value of 1.00. The learning rate of updating τ is set to 0.00001, so the timescales will not change too large. Gradient clipping is also used to prevent gradient explosion with a clipping value of 1.00 and learning rate is 2e-5. For regularization [4], a dropout of 0.5 was adopted on the LIMG to reduce overfitting. We use ALBERT to acquire word embeddings with the dimension of 128 and two heads in multi-head attention.

Competitor Methods

Model evaluation mainly focuses on two aspects, one of which is the pre-trained encoders. The texts encoding of LIMG incorporates labels information to highlight word vectors related to labels and it is necessary to verify the effectiveness of label-text fusion by comparing with word vectors without fused labels information. The second is the performance in long texts classification. In the experiments, we select excellent text classification models such as Char-CNN, Attn-LSTM, MTGRU and so on to analyze whether the performance of the improved GRU and CNN in long text classification is improved under the same input.

Experiment I: Comparative Accuracy Analysis of Classification Models

Experiment I compares the accuracy of LIMG and the baseline models on the four datasets above. Table 2 uses accuracy as a metric for classification and the equation is shown in Eq. (13). In order to comprehensively measure the performance of the models on the abstracts, Table 3 uses the class-weighted harmonic average $micro - F_1$ [33] to calculate the experimental results on two academic abstract datasets as shown in Eq. (14). $micro - F_1$ and $macro - F_1$ are commonly used to evaluate multi-classification tasks. $macro - F_1$, which calculates F_1 values for each category, is more susceptible to unbalanced data distribution than $micro - F_1$. Therefore, we choose $micro - F_1$ to evaluate classification performance.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{13}$$

$$micro - F_1 = \frac{\sum_{i=1}^{c} 2TP_i}{\sum_{i=1}^{c} 2TP_i + FP_i + FN_i} \tag{14}$$

The baseline models include the traditional classification model CNN, LSTM and their variants Attn-LSTM, Char-CNN, LSTM-CNN, Bi-LSTM and MTGRU. The pre-trained portion used by all baseline models uses word embeddings that fuse labels information.

From Table 3, LIMG has higher accuracy on all datasets than MTGRU, which is the best performing model in the baseline models. At the same time, LIMG achieves the maximum improvement of 2.28% on Amazon and Yah.A

Table 3 Accuracy of our model against other methods on various benchmark datasets

	AAPD	WOS	Amazon F	Amazon P	Yah.A
LSTM	67.80	74.31	59.43	93.90	70.84
Attn-LSTM	69.12	76.86	60.89	94.62	73.83
CNN	65.81	70.22	59.57	95.07	78.23
Char-CNN	63.12	71.51	59.57	95.23	71.20
LSTM-CNN	73.97	78.12	62.13	95.66	75.17
Bi-LSTM	70.36	77.02	61.11	95.54	72.62
MTGRU	74.51	78.57	65.18	95.87	78.86
Our model	**75.88**	**81.93**	**67.31**	**96.02**	**81.14**

The best results are in bold

datasets with long text lengths. This is because the hierarchical structure can obtain comprehensive features from different granularities of the texts.

Table 4 shows the $micro - F_1$ scores of each model on the AAPD and WOS datasets. It can be seen from Table 4 that LIMG has achieved a maximum improvement of 3.22% compared with MTGRU, which proves that the improved timescale can effectively filter out noise in abstracts and facilitate the extraction of fragmented distribution features.

The LIMG model has the best performance in two evaluation metrics, showing good generalization performance and can cope with various complex long text classification tasks.

Experiment II: Comparison with Large Pre-trained Models

Experiment II compares LIMG with several state-of-the-art pre-training models. Although some models use corpus of large-scale to get excellent language representation, it is difficult to learn specific meanings of labels in professional abstracts. The experiment uses $micro - F_1$ to measure the performance of these models in AAPD and WOS. The results are shown in Table 5.

As shown in Table 5, the $micro - F_1$ scores of the LIMG have improved 5.81 compared to other pre-trained models. By extracting the common semantics of similar samples, LIMG avoids the lack of actual semantics of labels. Therefore, the text representation ability is better than other models. The effect is most obvious on the WOS dataset, because the number of WOS labels is less than AAPD. Besides, there are more homogeneous samples for the model to learn and fewer labels are conducive to multi-head attention to pay more attention to the words that are related to the labels.

In Fig. 8, the assignment of weights in the attention layer is visualized, with different color treatments for the parts of the abstract that are relevant to different labels. The results show that the multi-head attention layer captures the label-related parts of the text sequence and verifies the effectiveness of the mechanism of fusing labels and information of abstracts.

Table 5 Comparison with pre-trained models in terms of micro-F1 scores

	AAPD	WOS
FastText	61.30	65.89
Word2vec	62.45	66.91
BERT	73.10	75.86
ALBERT	73.21	75.87
Our mode	**75.62**	**81.68**

The best results are in bold

Experiment III: Classification Performance on Different Length of Texts

In the text classification task, the accuracy of the model declines significantly due to the increase of text length. Therefore, experiment III divides the AAPD dataset according to different text lengths and evaluates them from six indicators: precious, recall, F1, micro-precious, micro-recall and micro – F_1 to measure the effect of the models on long texts comprehensively. The experiment has three parts. The first part compares whether GRU adds the classification indicators of the Timescale Shrink (TS), as shown in Fig. 9(a) and (b); the second part compares the classification performance before and after adding the DCP-CNN, as shown in Fig. 9(c) and (d); the third part compares LIMG with the optimal baseline model MTGRU at different text lengths, as shown in Fig. 9(e) and (f), the larger the area, the better the models perform.

As shown in Fig. 9(a) and (b), there is no large difference between the five indicators obtained by adding a TS to the same abstracts with a length of about 200. When processing abstracts with a length of about 400, the indicators of the model without TS decreased significantly and the model with TS decreased slightly. It indicates that TS can effectively avoid the above information forgetting and retain the long-term dependence of the context.

Table 4 Micro-F1 scores on the abstracts in AAPD and WOS datasets

	AAPD	WOS
LSTM	66.58	74.11
Attn-LSTM	68.77	76.53
CNN	65.01	69.82
Char-CNN	61.98	71.21
LSTM-CNN	73.14	77.10
Bi-LSTM	70.23	75.82
MTGRU	74.33	78.46
Our Model	**75.62**	**81.68**

The best results are in bold

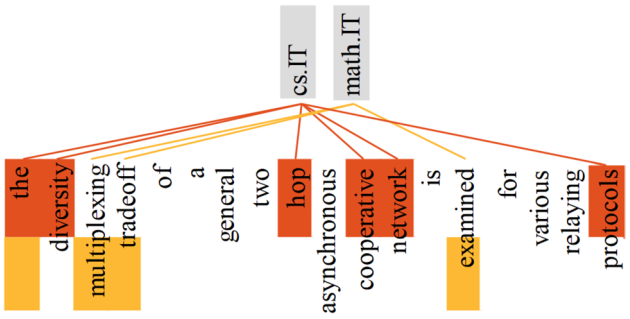
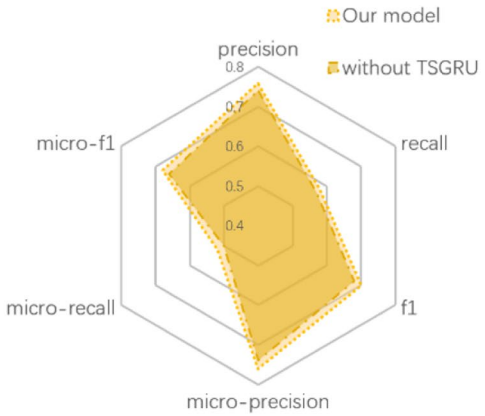
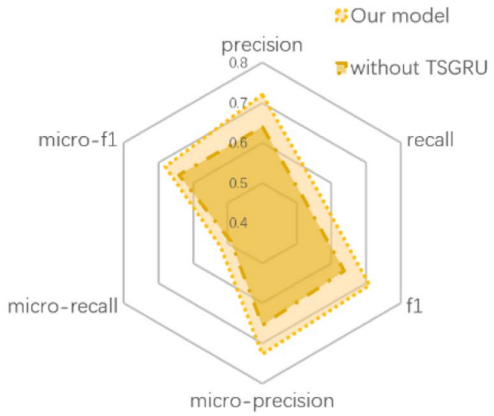


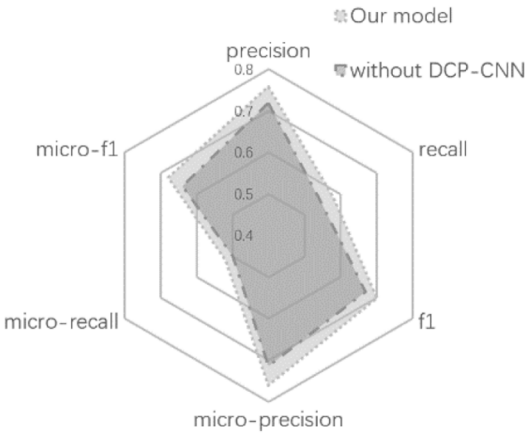
Fig. 8 Visualization of the attention scores in multi-head attention



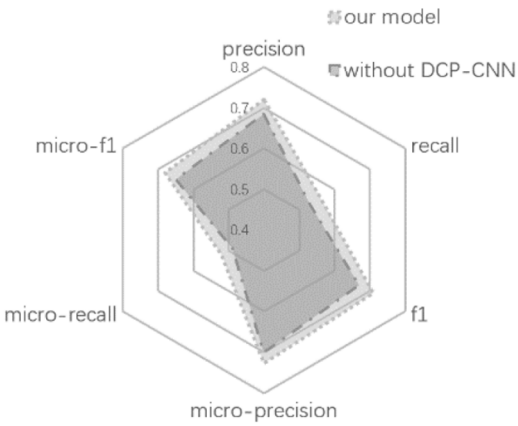
(a) 200 words



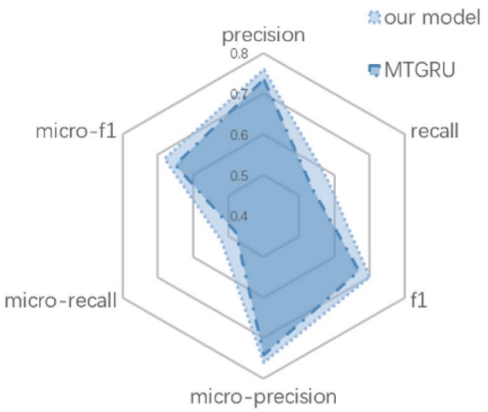
(b) 400 words



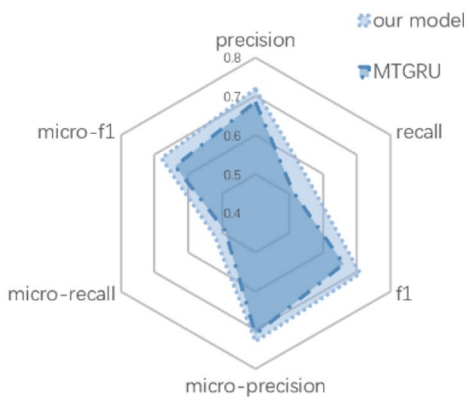
(c) 200 words



(d) 400 words

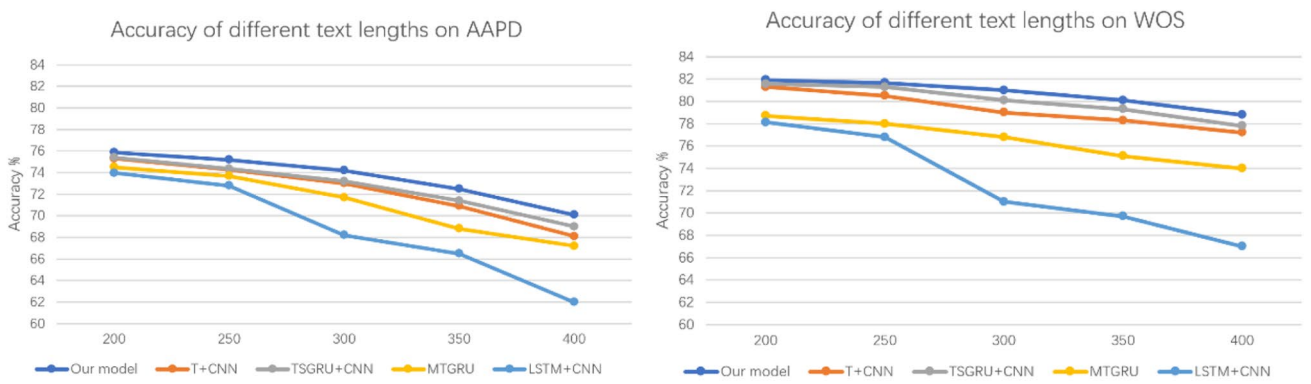


(e) 200 words



(f) 400 words

Fig. 9 Compare classification performance based on different lengths of input. TSGRU means timescale shrink GRU and DCP-CNN means dual channel pooling CNN. **a** 200 words, **b** 400 words, **c** 200 words, **d** 400 words, **e** 200 words, **f** 400 words



(a) AAPD dataset **(b) WOS dataset**

Fig. 10 Compare classification performance on AAPD and WOS datasets. TSGRU means timescale shrink GRU and T means timescale GRU. **a** AAPD dataset, **b** WOS dataset

In Fig. 9(c) and (d), the DCP-CNN-added model performs better, indicating that the dual channel pooling compensates the loss of key features caused by the GRU cell's special forgetting mechanism.

Figure 9(e) and (f) compares the indicators of LIMG with the optimal baseline model MTGRU. On datasets of different text lengths, LIMG outperforms MTGRU in all indicators.

Figure 10 further subdivides the text length and we can directly see the change of classification accuracy of each model as the text length increases. LSTM+CNN is the only model without adding timescale. Its accuracy decreases the most. Therefore, the timescale has the most significant improvement on long abstracts. The performance of TSGRU on shorter texts with soft thresholding algorithm is similar to that of ordinary timescale GRU. However, the gap between the two gradually widens with the increase of text length, which further illustrates the necessity of soft thresholding algorithm to filter text noise.

Experiment IV: Ablation Study

To further verify the effectiveness of the LIMG modules, Experiment IV conducts ablation studies on AAPD and WOS. Ablation studies usually refer to removing some features of a model or algorithm and observing how it affects model performance. The experiment is performed from the following three characteristics: Fusion Label Information Model (LI), Dual Channel Pooling Model (DCP), Shrink Time Scale Model (TS). Then calculate Accuracy and $micro - F_1$ scores on the datasets respectively, as shown in Tables 6 and 7.

Tables 6 and 7 show that the TS has the greatest impact on the overall performance of the model. TS introduces the past features after filtering noise to avoid the information being overwritten by the GRU and retains the long-range dependence. DCP and LI can also improve the performance of abstracts classification. The LI model explains that giving reasonable semantics to labels helps the model pay attention to the label-related features. While the DCP model extracts sentence-level features through a hierarchical structure, which is conducive to the distribution features of GRU aggregation fragmentation.

In order to demonstrate whether the features extracted by the model are beneficial to classification visually, the experiment uses the AAPD dataset to map the multi-dimensional features extracted by the model to the two-dimensional plane. We selected 5 labels that were not associated with each other randomly and packaged the abstracts belonging to these labels into a training set separately. The Principal Component Analysis (PCA) is used to map feature vectors to two-dimensional vectors. PCA can retain most of the feature information and avoid feature loss. We visualize the feature extraction results by using 5 colors to mark the abstracts of the 5 categories. Evaluate

Table 6 Accuracy of LIMG on AAPD and WOS datasets

LI	DCP	TS	AAPD	WOS
	✓	✓	74.77	79.19
✓			74.01	79.86
✓		✓	75.13	81.54
✓	✓	✓	75.88	81.93

Table 7 Micro-F1 scores of LIMG on AAPD and WOS datasets

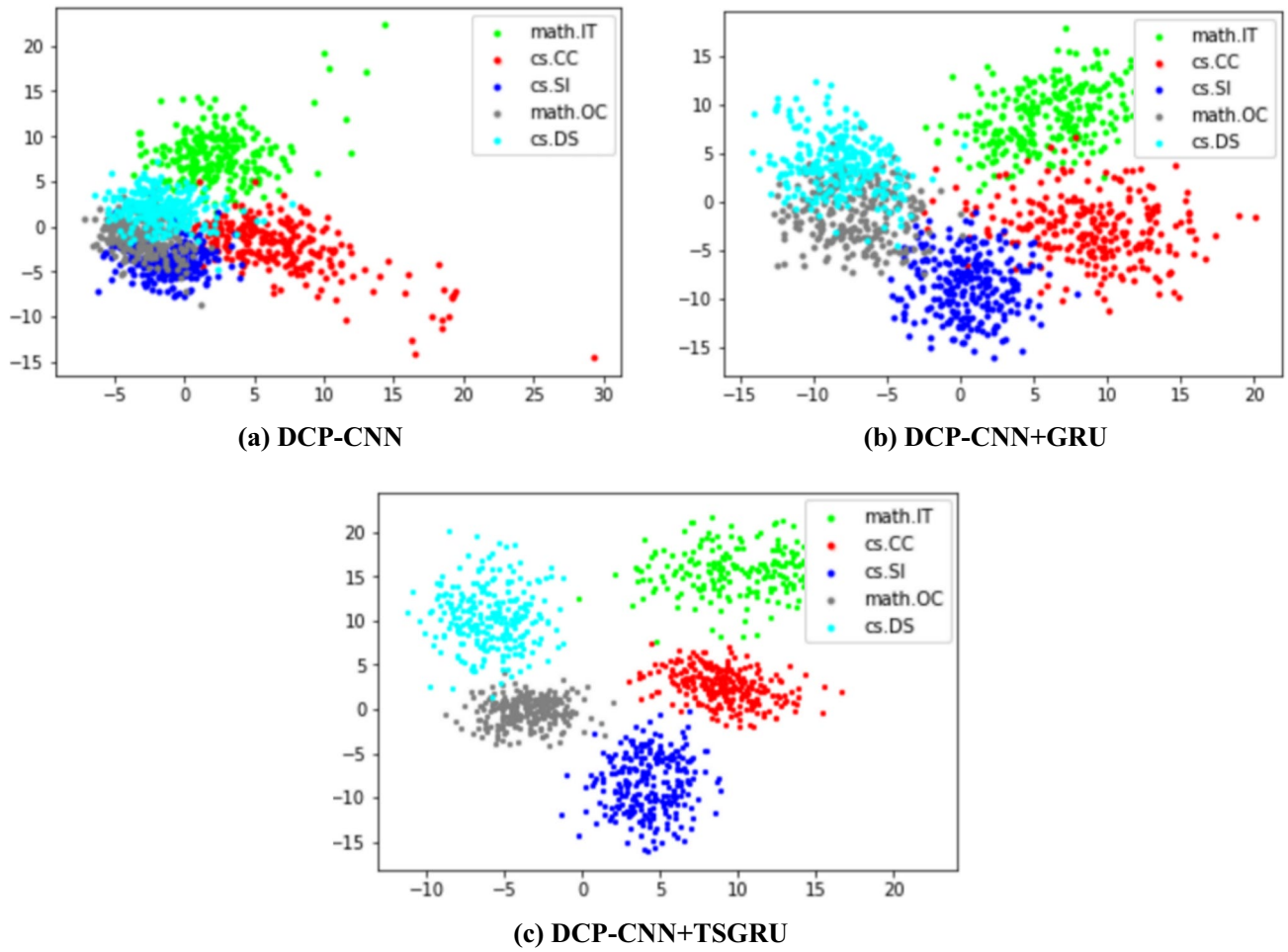
LI	DCP	TS	AAPD	WOS
	✓	✓	74.63	78.90
✓			73.89	79.53
✓		✓	74.96	81.31
✓	✓	✓	75.62	81.68

the results of extraction according to the degree of convergence of similar features and the boundary distance of heterogeneous features, as shown in Fig. 11.

In Fig. 11, TSGRU means timescale shrink GRU and DCP-CNN means dual channel pooling CNN. Figure 11(a) is a two-dimensional feature map of the dataset extracted by DCP-CNN, from which it can be seen that the boundary of various features is not obvious. Figure 11(b) further uses GRU to extract text information of different granularities on the basis of Fig. 11(a). It can be seen from the figure that the text characteristics of different categories have a relatively clear dividing line. Figure 11(c) is to add TS model on the basis of Fig. 11(b). Compared with Fig. 11(b), the dividing line of different features is more obvious and the degree of convergence is higher. Combined with the above three visual feature maps, it is shown that each part of the model has different contributions to the classification of abstracts.

Discussion

We compared the classification performance between baseline models and our model. As shown in Table 3 and 4, although the model is based on CNN and MTGRU [6], its performance has been significantly improved. The result of experiment III shows that TSGRU is particularly effective for processing the task of abstracts with long length. Because it filters out text noise while reducing the loss of information transmission in the deep network. This is similar to the purpose of residual networks used in image recognition [38]. This method helps computers process large amounts of information when simulating cognitive systems. It can be seen from Table 6 and Table 7 that the label vectors integrating text information also significantly improve the performance of classification. Assigning appropriate semantics to labels brings improvement for other cognitive domains. Just as in human



(a) DCP-CNN

(b) DCP-CNN+GRU

(c) DCP-CNN+TSGRU

Fig. 11 Visualize the features extracted by different models on AAPD dataset. The models extract the features of abstracts belonging to the five labels and we mark them with five colors. **a** DCP-CNN, **b** DCP-CNN+GRU, **c** DCP-CNN+TSGRU

cognition, labels contain some unique characteristics. For example, the polarity label used in the sentiment classification task [2, 17] is usually an integer. If the label contains the corresponding emotional information, the result may be improved. Since abstracts have a lot of volume and content, it is convenient to extract the corresponding label semantics. For data with sparsity and short text, the model has certain limitations. But with the help of external knowledge [32], this problem will be solved. Our model is suitable for all single-label and multi-label classification tasks with long texts, such as highly specialized abstracts and patent classification.

Conclusion

This paper discusses the problem of long text classification for abstracts. We develop a cognitively inspired multi-granularity long text classification model that integrates label information in view of the complex domain and the excessive length of abstracts. Firstly, the label information fusion model is designed to obtain the semantic information of each label to improve the semantic representation. Secondly, the dual channel pooling convolutional neural network (DCP-CNN) is proposed to solve the problem of loss of critical information due to excessive length of abstracts. Finally, the shallow semantic information channel in DCP-CNN and timescale shrink gated recurrent units (TSGRU) are used to obtain global information. On the basis of the timescale gated recurrent units, a soft threshold shrinkage algorithm is added to filter noise and enhance the long-term dependence in abstracts. In the experiments, the ablation studies are carried out on each part of the model. The results of the experiments show that the proposed model can maintain better performance with the gradual increase of length in abstracts. The model makes up for the shortcomings of the current classification models in the use of label semantics and its multi-granularity feature extraction solves both text noise and long-term dependency. As a result, computers can process large amounts of information in long abstracts, facilitating the cognitive system's *understanding* of academic texts. In the future, we plan to introduce external data to reduce the adverse effects of data sparseness on label information extraction and improve the encoding of academic terminology. This will allow us to improve the cognitive performance of the model.

Author Contribution Li Gao: Conceptualization, Methodology, validation, Formal Analysis, Writing, original draft preparation, Software, Writing—Reviewing and Editing, Supervision, Funding Acquisition. Yi Liu: Experiment, Data curation, Writing—Original draft preparation, Software, Writing—Reviewing and Editing, Supervision. Jianmin Zhu: Editing, Supervision. Zhen Yu: Writing—Reviewing and Editing, Supervision.

Funding The work was supported by the Ministry of Education Humanities and Social Sciences Foundation of China (20YJA870006). National Social Sciences Foundation of China (22BTQ021).

Data Availability The datasets used in this study are available to the public under a Creative Commons license: AAPD: https://git.uwaterloo.ca/jimmylin/Castor-data/tree/master/datasets/AAPD/data. WOS: https://paperswithcode.com/dataset/web-of-science-dataset. Amazom: https://snap.stanford.edu/data/web-Amazon.html. Yah.A: https://paperswithcode.com/dataset/yahoo-answers.

Declarations

Competing Interests The authors declare no competing interests.

References

1. Hahn M, Keller F. Modeling task effects in human reading with neural network-based attention. Cognition. 2023;230:105289.
2. Alatrash R, Priyadarshini R, Ezaldeen H, Alhinnawi A. Augmented language model with deep learning adaptation on sentiment analysis for E-learning recommendation. Cogn Syst Res. 2022;75:53–69.
3. Yun S, Cho W, Kim C. Technological trend mining: identifying new technology opportunities using patent semantic analysis. Inf Process Manage. 2022;59(4):102993.
4. Kaliyar RK, Goswami A, Narang P, Sinha S. FNDNet – a deep convolutional neural network for fake news detection. Cogn Syst Res. 2020;61:32–44.
5. Omolara AE, Alabdulatif A, Abiodun OI, Alawida M, Alabdulatif A, Alkhawaldeh RS. A systematic review of emerging feature selection optimization methods for optimal text classification: the present state and prospective opportunities. Neural Comput Appl. 2021;33:15091–118.
6. Moirangthem DS, Lee M. Hierarchical and lateral multiple timescales gated recurrent units with pre-trained encoder for long text classification. Expert Syst Appl. 2021;165:113898.
7. Tan C, Ren Y, Wang C. An adaptive convolution with label embedding for text classification. Appl Intell. 2022;53:804–12.
8. Asudani DS, Nagwani NK, Singh P. Impact of word embedding models on text analytics in deep learning environment: a review. Artif Intell Rev. 2023;56:10345–425. https://doi.org/10.1007/s10462-023-10419-1.
9. Zia S, Azhar M, Lee B, Tahir A, Ferzund J, Murtaza F, et al. Recognition of printed Urdu script in Nastaleeq font by using CNN-BiGRU-GRU based encoder-decoder framework. Intelligent Systems with Applications. 2023;18:200194.
10. Cao H, Zhao T, Wang W, Wei P. Bilingual word embedding fusion for robust unsupervised bilingual lexicon induction. Information Fusion. 2023;97:101818.
11. Mahto D, Yadav S C. Emotion prediction for textual data using GloVe based HeBi-CuDNNLSTM model. Multim Tools Appl. 2023.
12. Tagarelli A, Simeri A. Unsupervised law article mining based on deep pre-trained language representation models with application to the Italian civil code. Artificial Intelligence and Law. 2021;30:417–73.
13. Chen C, Wang K, Hsiao Y, Chou J. ALBERT: an automatic learning based execution and resource management system for optimizing Hadoop workload in clouds. Journal of Parallel and Distributed Computing. 2022;168:45–56.

14. Hassan SU, Ahamed J, Ahmad K. Analytics of machine learning-based algorithms for text classification. Sustainable Operations and Computers. 2022;3:238–48.
15. Pavan Kumar RK, Jayagopal P. Context-sensitive lexicon for imbalanced text sentiment classification using bidirectional LSTM. J Intell Manuf. 2021;34:2123–32.
16. Huang Y, Liu Q, Peng H, Wang J, Yang Q, Orellana-Martín D. Sentiment classification using bidirectional LSTM-SNP model and attention mechanism. Expert Syst Appl. 2023;221:119730.
17. Zhang Y, Tiwari P, Song D, Mao X, Wang P, Li X, et al. Learning interaction dynamics with an interactive LSTM for conversational sentiment analysis. Neural Netw. 2021;133:40–56.
18. Huan H, Guo Z, Tingting C, He Z. A text classification method based on a convolutional and bidirectional long short-term memory model. Connect Sci. 2022;34(1):2108–24.
19. Lu G, Liu Y, Wang J, Wu H. CNN-BiLSTM-Attention: a multi-label neural classifier for short texts with a small set of labels. Inf Process Manage. 2023;60(3):103320.
20. Luo T, Liu Y, Li T. A multi-feature fusion method with attention mechanism for long text classification. 2022 the 6th International Conference on Compute and Data Analysis. 2022.
21. Kenarang A, Farahani M, Manthouri M. BiGRU attention capsule neural network for Persian text classification. J Ambient Intell Humaniz Comput. 2022;13:3923–33.
22. Yang S, Wang J, Zhang N, Deng B, Li X. Azghadi M R Cerebellu Morphic: large-scale neuromorphic model and architecture for supervised motor learning. IEEE Trans Neural Netw Learn Syst. 2021;33(9):4398–412.
23. Yang S, Wang J, Hao X, Li H, Wei X, Deng B, Loparo KA. BiCoSS: Toward large-scale cognition brain with multigranular neuromorphic architecture. IEEE Trans Neural Netw Learn Syst. 2021;33(7):2801–15.
24. Pal A, Singh KP. AdamR-GRUs: Adaptive momentum-based regularized GRU for HMER problems. Appl Soft Comput. 2023;143:110457.
25. Aote SS, Pimpalshende A, Potnurwar A, Lohi S. Binary particle swarm optimization with an improved genetic algorithm to solve multi-document text summarization problem of Hindi documents. Eng Appl Artif Intell. 2023;117:105575.
26. Herrera A, Sánchez N G, Vargas D. Rule-based Spanish multiple question reformulation and their classification using a convolutional neuronal network. Comput Sist. 2021;25(1).
27. Kaur K, Kaur P. BERT-CNN: improving BERT for requirements classification using CNN. Procedia Computer Science. 2023;218:2604–11.
28. Rafiepour M, Sartakhti JS. CTRAN: CNN-transformer-based network for natural language understanding. Eng Appl Artif Intell. 2023;126:107013.
29. Liang Y, Li H, Guo B, Yu Z, Zheng X, Samtani S, Zeng D. Fusion of heterogeneous attention mechanisms in multi-view convolutional neural network for text classification. Inf Sci. 2021;548:295–312.
30. Ayetiran EF. Attention-based aspect sentiment classification using enhanced learning through CNN-BiLSTM networks. Knowl Based Syst. 2022;252:109409.
31. Ahmed Z, Wang J. A fine-grained deep learning model using embedded-CNN with BiLSTM for exploiting product sentiments. Alex Eng J. 2022;65:731–47.
32. Li H, Yan Y, Wang S, Liu J, Cui Y. Text classification on heterogeneous information network via enhanced GCN and knowledge. Neural Comput Appl. 2023;35:14911–27.
33. Liu M, Liu L, Cao J, Du Q. Co-attention network with label embedding for text classification. Neurocomputing. 2022;471:61–9.
34. Wang J, Chen Z, Qin Y, He D, Lin F. Multi-aspect co-attentional collaborative filtering for extreme multi-label text classification. Knowledge Based Systems. 2022;260:110110.
35. Qian T, Li F, Zhang M, Jin G, Fan P, Wenhua D. Contrastive learning from label distribution: a case study on text classification. Neurocomputing. 2022;507:208–20.
36. Wang Q, Zhu J, Shu H, Asamoah KO, Shi J, Zhou C. GUDN: a novel guide network with label reinforcement strategy for extreme multi-label text classification. J King Saud Univ Comput Inf Sci. 2023. https://doi.org/10.1016/j.jksuci.2023.03.009.
37. Su L, Xiong L, Yang J. Multi-Attn BLS: Multi-head attention mechanism with broad learning system for chaotic time series prediction. Appl Soft Comput. 2023;132:109831.
38. Joshi A, Hong Y. R2Net: Efficient and flexible diffeomorphic image registration using Lipschitz continuous residual networks. Med Image Anal. 2023;89:102917.