Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.048135

ARTICLE

Enhancing Deep Learning Semantics: The Diffusion Sampling and Label-Driven Co-Attention Approach

Chunhua Wang[1,2], Wenqian Shang[1,2,*], Tong Yi[3,*] and Haibin Zhu[4]

[1]State Key Laboratory of Media Convergence and Communication, Communication University of China, Beijing, 100024, China

[2]School of Computer and Cyber Sciences, Communication University of China, Beijing, 100024, China

[3]School of Computer Science and Engineering, Guangxi Normal University, Guilin, 541004, China

[4]Department of Computer Science, Nipissing University, North Bay, ON P1B 8L7, Canada

*Corresponding Authors: Wenqian Shang. Email: shangwenqian@cuc.edu.cn; Tong Yi. Email: yitong@mailbox.gxnu.edu.cn

Received: 28 November 2023 Accepted: 12 March 2024 Published: 15 May 2024

ABSTRACT

The advent of self-attention mechanisms within Transformer models has significantly propelled the advancement of deep learning algorithms, yielding outstanding achievements across diverse domains. Nonetheless, self-attention mechanisms falter when applied to datasets with intricate semantic content and extensive dependency structures. In response, this paper introduces a Diffusion Sampling and Label-Driven Co-attention Neural Network (DSLD), which adopts a diffusion sampling method to capture more comprehensive semantic information of the data. Additionally, the model leverages the joint correlation information of labels and data to introduce the computation of text representation, correcting semantic representation biases in the data, and increasing the accuracy of semantic representation. Ultimately, the model computes the corresponding classification results by synthesizing these rich data semantic representations. Experiments on seven benchmark datasets show that our proposed model achieves competitive results compared to state-of-the-art methods.

KEYWORDS

Semantic representation; sampling attention; label-driven co-attention; attention mechanisms

1 Introduction

The Transformer model has introduced a groundbreaking paradigm in deep learning architectures, leading to considerable advancements in a multitude of tasks [1,2]. The core of the Transformer architecture is the self-attention mechanism, a pivotal innovation in model architecture. This mechanism dynamically isolates essential features and substantially enhances model performance across numerous natural language processing (NLP) domains, thereby becoming an integral component of contemporary deep learning frameworks [3].

The computational structure of self-attention, as proposed in the Transformer model by Vaswani et al., merges the benefits of context feature aggregation with the parallel computation of essential attributes. This enables the dynamic NLP capture of pertinent information within the data, leading to

significant improvements across a variety of tasks [4]. However, the models' computational efficiency when managing large-scale data dependencies during training remains a challenge. Addressing this, Dai et al. proposed an innovative approach that merges block recursion with relative positional encoding to enhance computational performance [5]. To more efficiently extract fundamental semantic elements from text, Zhou et al. integrated an attention mechanism with a Bidirectional Long Short Term Memory (Bi-LSTM) network, resulting in a transparent architecture that delineates each word's contribution to the model's inferential outcomes [6].

Despite extensive refinements to the self-attention mechanism within deep learning models by various researchers, models that depend exclusively on this mechanism frequently fail to adequately capture the intricate nuances of context present in extensive text corpora. Consequently, this limitation hampers their efficiency in handling lengthy sequences [7]. Additionally, the word vectors utilized by self-attention for semantic analysis are typically derived from broad, open datasets. While this training approach enhances the versatility of word vectors across a range of domains and tasks, it also inadvertently reduces the precision of semantic representations needed for specialized applications [8,9]. Such a trade-off between versatility and specificity detrimentally affects the model's proficiency in detailed semantic interpretation [8].

To enhance the semantic representation capabilities of the self-attention mechanism and alleviate issues of semantic inaccuracies caused by single input sources, Wang et al. introduced a label embedding framework that incorporates label information during the model's computation process to improve the computational efficiency of textual semantics [10]. Nonetheless, label information's direct contribution to the decision-making phase provides only an indirect influence on the semantic representation learning process. Liu et al. addressed these limitations by employing deep canonical correlation techniques to couple label and textual semantic information, which enhanced model performance but also introduced complexity and computational demands [11]. In response, Liu et al. devised a model structure for collaborative encoding of texts and labels, which succeeded in capturing their prominent features and delivering superior text classification results [12]. While these methods have made progress in the integrated representation of text and labels, they have not fully addressed the issues stemming from inherent data biases and context sensitivity, nor have they fundamentally improved the intrinsic shortcomings of the self-attention mechanism in capturing coherent and deep semantic structures.

Considering previous research and after a thorough analysis of the challenges existing models face in computing data's semantic representation, and inspired by the brain's strategy for processing textual information—a cognitive shift from global to detailed understanding [13]. To address these issues, we introduce the DSLD, an advanced deep learning framework. This framework is designed to synergize context embedding strategies with self-supervised learning processes, thereby effectively narrowing the gap between detailed focus on specific features and the capture of broad contextual semantics. The DSLD architecture aims to mitigate data's inherent biases and increase sensitivity to contextual variations while enhancing the self-attention mechanism's ability to comprehend coherent and hierarchically rich semantic structures.

In this paper, we propose the DSLD model, which features two principal components: The Diffusion Sampling Encoder and the Label-Driven Encoder. The Diffusion Sampling Encoder incorporates a novel multi-channel diffusion sampling convolutional attention mechanism that transcends conventional attention models. It enhances semantic precision by distributing focus across various information channels, thus providing a more nuanced representation of data semantics. This advancement not only expands the model's semantic comprehension but also improves its flexibility

in adapting to complex data configurations. The Label-Driven Encoder utilizes a targeted learning approach, directing text representation with data labels and seamlessly melding label information into the attention process. This integration grants precise control over label-specific features within semantic learning, markedly refining the model's semantic accuracy. Such depth of integration propels the model beyond mere label identification towards a more profound semantic insight, facilitating a smooth transition from ambiguous to definitive text understanding. The innovative DSLD model approach surpasses the constraints inherent in conventional methodologies and extant attention mechanisms, offering a more nuanced and comprehensive linguistic semantic interpretation for text classification tasks. Extensive evaluations conducted on seven benchmark datasets substantiate the efficacy of the DSLD framework.

For this paper, the main contributions are as follows:

1. To heighten the model's responsiveness to variances in contextual semantics during text computation, we have developed a multi-channel diffusion sampling attention computation framework. Diverging from conventional self-attention mechanisms, this approach effectively captures a more nuanced spectrum of semantic representations within texts of differing informational densities. It also exhibits a stronger capacity for processing extensive semantic dependencies within the text.

2. To reduce the intrinsic bias in data representation induced by open training and improve the model's ability to process comprehensive data semantics, we have devised a label-driven attention computation strategy. This novel approach exploits the correlation between labels and data to enable input data to assimilate label information, thereby diminishing representational bias. Concurrently, incorporating label information into the semantic representation learning process of the data bolsters the accuracy of the model's semantic calculations.

3. We have designed and implemented the DSLD neural network model, incorporating a diffusion sampling encoder and label-driven encoder, and demonstrated its superior performance over other deep learning text classification methods on seven benchmark datasets.

The remainder of the paper is organized as follows: Section 2 provides a literature review of deep learning models utilizing attention mechanisms and label embedding methods in text classification tasks. Section 3 offers a comprehensive introduction to the proposed DSLD model. Section 4 presents the experimental results, comparing the proposed model against various baseline models to assess its performance. Finally, the paper concludes with a summary.

2 Related Work

2.1 Attention Mechanism

The pursuit of accurate semantic representation in text classification contends with semantic complexity and high-dimensional, sparse text data. Among various approaches, the attention mechanism has been a critical advancement, enabling deep learning models to concentrate on salient features during inference, and has gained traction in text classification research.

As depicted in Fig. 1, the attention mechanism functions as an intricate selection process. It computes attention scores by contrasting a query with corresponding keys, normalizes these scores, and applies the resultant attention weights to values, producing the final attention output. This method excels in its targeted processing, dynamically allocating computational focus to information pertinent to the current neural network task, avoiding the exhaustive processing of all inputs.

Beyond the self-attention structure previously described, the application of soft attention techniques has made significant advances in the calculation of contextual semantics for input sequences

[14]. The efficacy of these mechanisms is attributed to the formulation of context vectors, which are computed as a weighted sum of all hidden states within the sequence. Liu et al. furthered the progress in this domain by introducing a hybrid model that amalgamates multi-stage attention with Temporal Convolutional Networks (TCN) and Convolutional Neural Networks (CNN). This model not only elevates computational parallelism by leveraging TCN but also accentuates stage-specific discriminative features through the strategic integration of attention within various CNN layers, thus significantly refining model precision [15]. To address the challenges posed by incomplete information more capably, Chen et al. developed a neural network architecture that synthesizes semantic priors with profound attention residual groups. The architecture employs these semantic priors to infer missing information via attention mechanisms [16].

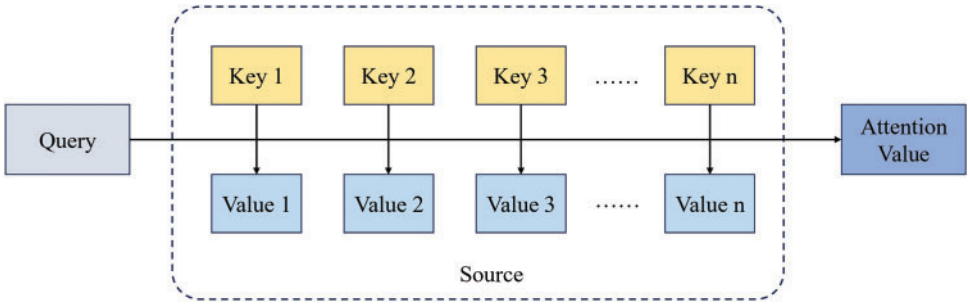

Figure 1: The processing flow of attention mechanism data

Previous research has predominantly centered on improving the precision of semantic information computation by layering numerous attention levels and amalgamating multiple model structures. In contrast, our study introduces a model architecture capable of deriving richer semantic information through a singular attention layer, harnessing a multi-channel diffusion sampling technique. This method affords the model a comprehensive and detailed semantic grasp of text data.

2.2 Label Embedding

Label embedding, a technique that incorporates label information into models, has proven markedly effective across various domains [17–19]. This approach facilitates the enhancement of model performance by utilizing label information in data inference. Following its successful implementation in image processing, exploration of label information's applications in natural language processing has commenced. Tang et al. broke new ground with semi-supervised learning in a text embedding model that employs both labeled and unlabeled data, thereby mapping this information within expansive heterogeneous text networks [20]. In the multi-task learning domain, Zhang et al. leveraged semantic task correlations to develop a multi-task label embedding model, recasting classification as a vector matching endeavor and thus fortifying label semantic representations while mitigating the information loss associated with independent task labels [21]. Nonetheless, earlier investigations concentrated exclusively on the computational relationships between labels and data, overlooking the intricate interrelations among labels. To rectify this, Pappas et al. introduced a joint input label model, enhancing and surmounting the constraints of analogous antecedent models [22]. Despite this progress, label information has typically been embedded separately from feature computation in extant research. Furthering this work, Liu et al. crafted a semantic computation framework integrating label embedding with bidirectional attention to discern the semantic interplay between granular token text representations and label embeddings, thereby refining the model's computational

efficacy [23]. Responding to earlier research that zeroed in on extracting distinct text representations, Liu et al. envisioned a collaborative attention network that incorporates label embedding, allowing the model to account for pertinent interrelations between labels and data [12].

Despite the commendable achievements of prior research, it has neglected semantic biases that arise when input data, trained in open domains, undergo semantic representation computation. To address this shortcoming and achieve a deeper integration of label information with data semantics, this study presents an innovative label embedding technique. This method augments the representation of the original data by determining the combined semantic information of the text and its associated labels. Subsequently, it fuses this label information with the data's semantic computations through meticulous attention-based processes. This strategy results in a precise inclusion of label semantics into the computations of text representation, thus enhancing the model's classification precision.

3 Diffusion Sampling Label-Driven Neural Network

In this study, we introduce the DSLD model, which performs comprehensive multi-layered semantic computations of input contexts while integrating label information for joint semantic analysis. The model's architecture is depicted in Fig. 2. As the data is fed into the model, it is concurrently processed by the Diffusion Sampling Encoder and the Label-driven Encoder, performing semantic analyses across various stratifications and from a holistic to a detailed approach, respectively. Ultimately, the classification layer amalgamates the computed semantic vectors to ascertain the likelihood that the input corresponds to each category.

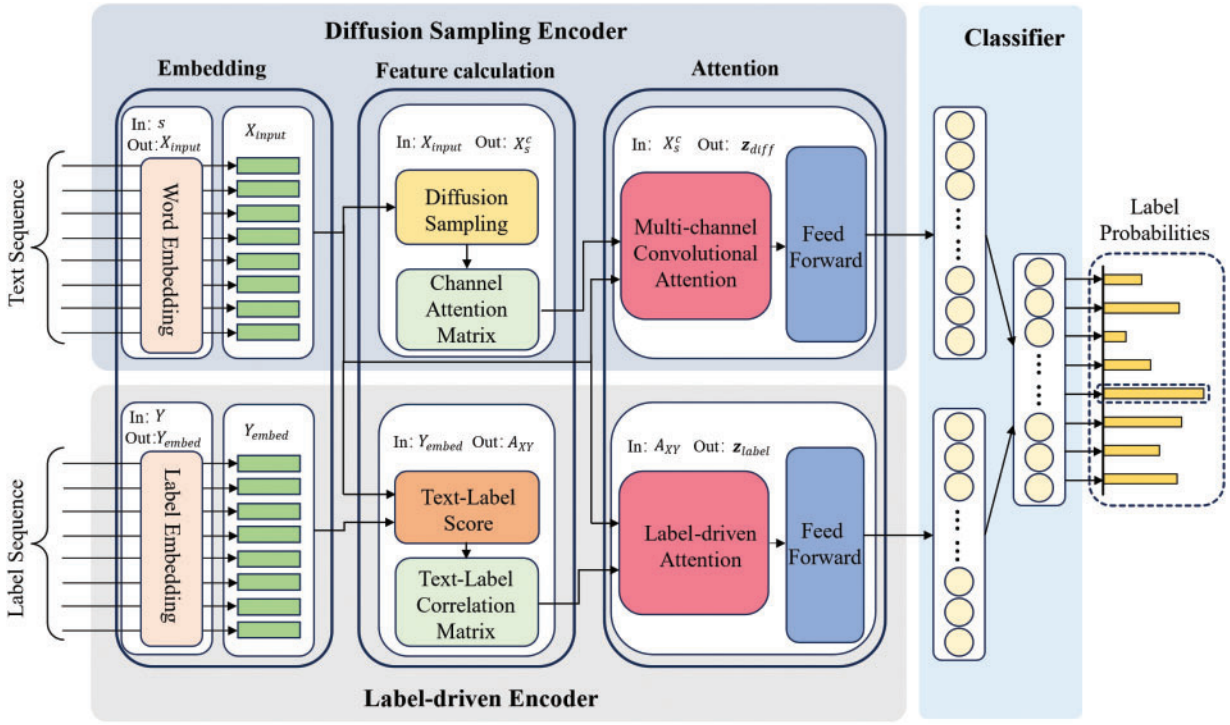

Figure 2: The architectural diagram of the DSLD model

The model consists of two primary components:

(1) Diffusion Sampling Encoder, which, unlike conventional attention-based methods that directly process data, employs a multi-channel diffusion sampling method to capture contextual semantic information more exhaustively by gathering related information matrices across different information densities. Following this, convolutional attention synthesizes the matrices to produce semantic representation vectors using attention mechanisms.

(2) Label-driven Encoder, recognizing that label information is instrumental in semantic computations and provides a significant direction for semantic representation. This component calibrates the representation of data by aligning it with the corresponding label information, thus addressing the semantic biases that may result from the training of data in open domains. Moreover, it incorporates label information into the semantic computation process via the attention mechanism to bolster computational precision.

3.1 Diffusion Sampling Encoder

Effective text classification relies on contextual information, which constitutes a crucial semantic component in textual data. Therefore, this study integrates an attention mechanism to calculate the features of the input data. In contrast to conventional attention-based methods, this study employs a multi-channel diffusion sampling approach to capture contextual semantic information more comprehensively by acquiring relevant information matrices at multiple information densities. Subsequently, these matrices undergo synthesis through the use of convolutional attention, resulting in the computation of the semantic representation vector for the input data.

3.1.1 Diffusion Sampling

This paper introduces the diffusion sampling method to enhance semantic information extraction by sampling the original data and generating datasets with diverse information densities [24]. This enables the utilization of attention mechanisms for computing semantic information across varied data densities.

In a formal manner, for an input sequence s consisting of m word tokens, after being represented by a word vector encoder with encoding dimension d, a numerical matrix $\boldsymbol{X}_{input} \in \Re^{m\times d}$ is obtained. Subsequently, a sampling matrix $\boldsymbol{M} \in \Re^{m\times d}$, of the same dimensionality, is generated to correspond to $\boldsymbol{X}_{input}$. The parameters within the sampling matrix M follow a binomial distribution with parameters n and p, denoted as $\boldsymbol{M} \sim B(n,p)$. Here, n being equal to 1 implies that the parameters are sampled only once, and p represents the probability of parameter sampling. In this case, the expected value $E(\boldsymbol{M})$ of the sampling matrix $\boldsymbol{M}$ is equal to p. The sampled numerical matrix $\boldsymbol{X}_s$ is obtained by performing element-wise Hadamard product (denoted as "$\odot$") between the numerical matrix $\boldsymbol{X}_{input}$ and the sampling matrix $\boldsymbol{M}$, as shown in Eq. (1).

$$\boldsymbol{X}_s = \boldsymbol{X}_{input} \odot \boldsymbol{M} \tag{1}$$

By progressively stacking the sampling matrices, one can obtain the numerical matrix X_s^t at the corresponding iteration step t. The computational procedure is detailed as shown in Eq. (2).

$$\boldsymbol{X}_s^t = \boldsymbol{X}_s^{t-1} \odot \boldsymbol{M} \tag{2}$$

When the initial information density, denoted as $S(\boldsymbol{X}_{input})$, of the original numerical matrix is set to 1, the subsequent information density, $S(\boldsymbol{X}_s^t)$, across various iteration steps is shown in Eq. (3).

$$S\left(\boldsymbol{X}_s^t\right) = S\left(\boldsymbol{X}_s^{t-1}\right) * E\left(\boldsymbol{M}\right) = 1 * E\left(\boldsymbol{M}\right)^t = p^t \tag{3}$$

The introduction of the sampling matrix $\boldsymbol{M}$ enables the generation of numerical matrices with varying information densities across different iteration steps. To reduce the dependence of linear diffusion sampling methods on the sampling order, this paper further proposes a multi-channel sampling calculation method. This method adopts a parallel approach while conducting diffusion sampling on the data through multiple channels, as shown in Fig. 3. The calculation process of the sampled value matrix $\boldsymbol{X}_s^c$ in each channel is shown in Eq. (4).

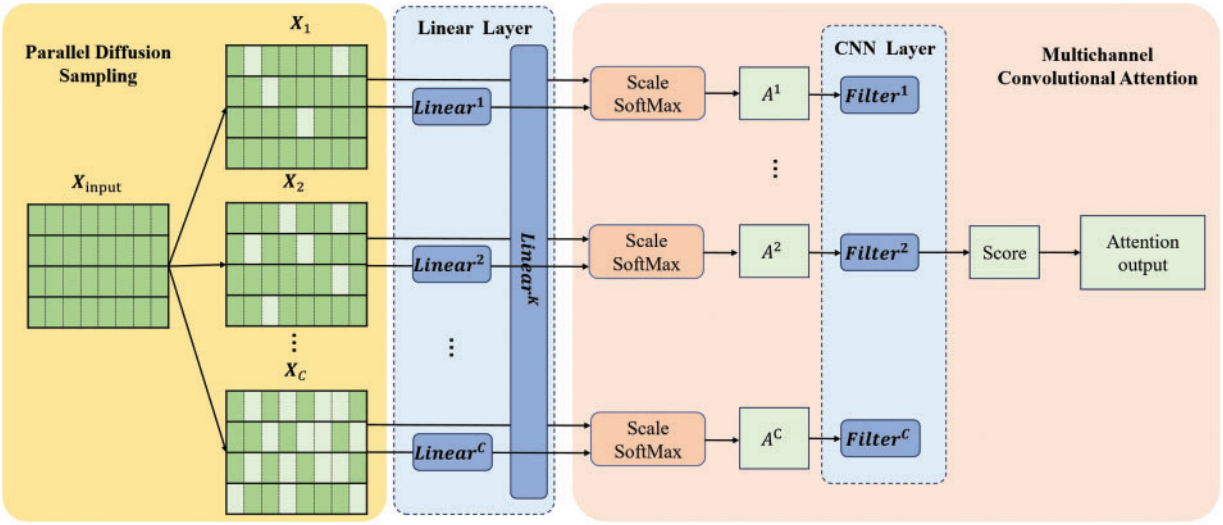

Figure 3: Calculation flow of parallel diffusion sampling and multichannel convolutional attention

$$\boldsymbol{X}_s^c = \boldsymbol{X}_{input} \odot \boldsymbol{M}^c \tag{4}$$

In the formula, $\boldsymbol{M}^c$ represents a sampling matrix with varying sampling probabilities in different channels. The parameters within the sampling matrix follow a binomial distribution with parameters n and p^c, denoted as $\boldsymbol{M}^c \sim B\left(n, p^c\right)$. Here, when n equals 1, it signifies that the parameters are sampled only once at random, while p^c represents the probability of parameter sampling within channel c. The calculation process for p^c is shown in Eq. (5).

$$p^c = \alpha\left(\frac{C_{num} - c}{C}\right) + \beta \tag{5}$$

In this formula, C_{num} signifies the total channel count, c indicates the current channel index, while α and β comprise a set of hyperparameters responsible for governing the channel's sampling probability within the interval [0, 1].

Following enhancement, the refined diffusion sampling process guarantees the mutual independence of information density computation across channels, eliminating reliance on sequential order. Assuming the information density of the original numerical matrix, denoted as $S\left(\boldsymbol{X}_{input}\right)$, equals 1, we calculate the information density of distinct channel-specific numerical matrices, $S(\boldsymbol{X}_s^c)$, as

shown in Eq. (6).

$$S\left(\boldsymbol{X}_s^c\right) = S\left(\boldsymbol{X}_{input}\right) * E\left(\boldsymbol{M}^c\right) = 1 * p^c = p^c \tag{6}$$

3.1.2 Multichannel Convolutional Attention

After applying the previously mentioned diffusion sampling method, we can acquire numerical matrices labeled as $\boldsymbol{X}_s^c$ with diverse information densities. Next, once we have these numerical matrices, we can calculate the semantic information embedded in the data by employing attention mechanisms. In contrast to the traditional approach of conducting attention computations on raw data, the utilization of diffusion-sampled data manifests varying information densities, thereby facilitating the extraction of more comprehensive semantic information.

In conventional attention mechanisms, before computation, input data undergoes three independent linear transformations, resulting in new data representations $\boldsymbol{Q} \in \Re^{q \times d_q}$, $\boldsymbol{K} \in \Re^{k \times d_k}$, and $\boldsymbol{V} \in \Re^{v \times d_v}$. Typically, $q = k = v$, and $d_q = d_k = d_v$. The calculation of attention consists of two primary steps. First, we compute the attention matrix $\boldsymbol{A}$ using $\boldsymbol{Q}$ and $\boldsymbol{K}$, as shown in Eq. (7). Then, we obtain the final attention output by matrix-multiplying the attention matrix $\boldsymbol{A}$ with $\boldsymbol{V}$, as shown in Eq. (8).

$$\boldsymbol{A} = softmax\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_k}}\right) \tag{7}$$

$$Attention = AV \tag{8}$$

In this paper, we propose a multi-channel convolutional attention computation method tailored for analyzing multi-channel data acquired via diffusion sampling. We utilize this method to calculate attention matrices, denoted as $\boldsymbol{A}^c$, for various data channels. Concurrently, it aggregates these channel-specific attention matrices into a unified attention matrix, denoted as $\boldsymbol{A}^{count}$, employing convolutional techniques. Subsequently, it utilizes the merged attention matrix and input data to compute information weighted by attention relationships. The procedure for obtaining attention matrices, denoted as $\boldsymbol{A}^c$, across various data channels, is shown in Eqs. (9) to (11).

$$\boldsymbol{Q}^c = Linear^c\left(\boldsymbol{X}_{mask}^c\right) \tag{9}$$

$$\boldsymbol{K}^c = Linear^K\left(\boldsymbol{X}_{mask}^c\right) \tag{10}$$

$$\boldsymbol{A}^c = softmax\left(\frac{\boldsymbol{Q}^c\boldsymbol{K}^{c\,T}}{\sqrt{d_k}}\right) \tag{11}$$

In the formula, $\boldsymbol{Q}^c \in \Re^{q \times d_q}$ represents the new representation of the data obtained by computing the numerical matrix $\boldsymbol{X}_s^c$ sampled from channel c through the corresponding linear layer $Linear^c$, where c denotes the channel index. Similarly, $\boldsymbol{K}^c \in \Re^{k \times d_k}$ represents the new data representation obtained by computing the numerical matrix $\boldsymbol{X}_s^c$ sampled from channel c through the linear layer $Linear^K$. It is important to note that the linear layer $Linear^K$ is shared across different data channels. Here, d_k denotes the data dimensionality of $\boldsymbol{K}^c$. In this context, $\boldsymbol{Q}^c \in \Re^{q \times d_q}$, $\boldsymbol{K}^c \in \Re^{k \times d_k}$, and $\boldsymbol{V} \in \Re^{v \times d_v}$ all have the same dimensions, where $q = k = v = m$, and $d_q = d_k = d_v = d$.

The attention matrix $\boldsymbol{A}^{c} \in \Re^{m\times d}$ After that, a comprehensive attention matrix $\boldsymbol{A}^{count} \in \Re^{m\times d}$. The calculation process is shown in Eq. (12).

$$\boldsymbol{A}_{i,j}^{count} = \sigma\left(\sum\nolimits_{c}^{C}\sum\nolimits_{w}^{K_W}\sum\nolimits_{h}^{K_H}\boldsymbol{A}_{i+w,j+h,c}\cdot\boldsymbol{F}_{w,h,c}+b\right) \tag{12}$$

In the formula, i and j denote the positional coordinates of parameters within the matrix, $\sigma(\cdot)$ represents the activation function, C signifies the number of channels, K_W represents the width of the convolutional kernel, K_H represents the height of the convolutional kernel, $\boldsymbol{F} \in \Re^{K_W\times K_H\times C}$ denotes the convolutional kernel, and b denotes the bias term corresponding to the convolutional kernel.

The comprehensive attention matrix, denoted as $\boldsymbol{A}^{count}$, obtained through convolutional operations, and the representation of the original data processed through a linear layer, as shown in Eq. (13), are combined to compute the output of multi-channel convolutional attention, denoted as $Attention_{CC}\left(\boldsymbol{X}_{input}\right)$, as shown in Eq. (14).

$$\boldsymbol{V} = Linear^{V}\left(\boldsymbol{X}_{input}\right) \tag{13}$$

$$Attention_{CC}\left(\boldsymbol{X}_{input}\right) = \boldsymbol{A}^{count}\boldsymbol{V} \tag{14}$$

As previously elucidated, the diffusion sampling encoder leverages the diffusion sampling technique for gathering textual data with diverse information densities. It subsequently conducts a thorough computation of semantic information using a multi-channel convolutional attention mechanism.

3.2 Label-Driven Encoder

This study endeavors to correct semantic bias in data representation and effectively incorporate label information into semantic computation by proposing a label-driven encoding methodology. Initially, the approach encodes the label information, following which an attention mechanism computes attention scores to assess the relevance of the input text to each label. Subsequently, these scores are harnessed through attention-based methods to facilitate the feature extraction process for the input text, culminating in the production of a representative vector for the text.

To encode meaningful correlations between labels and input data, this study introduces an attention mechanism for computing the relevance between context within the text and the associated labels. For a given input word sequence $s \in \Re^{m}$ and a sequence of task labels $\boldsymbol{Y} \in \Re^{L}$, they are initially mapped into word embedding sequences $\boldsymbol{X}_{input} \in \Re^{m\times d}$ and label embedding sequences $\boldsymbol{Y}_{embed} \in \Re^{L\times d_Y}$, respectively. Subsequently, a trainable weight matrix $\boldsymbol{W}_X \in \Re^{d\times d_Y}$ is utilized to compute a scoring matrix $\boldsymbol{X}_{Score} \in \Re^{m\times d_Y}$ representing the relevance between the data and labels, as shown in Eq. (15). Based on the scoring matrix $\boldsymbol{X}_{Score}$ and the label embedding sequence $\boldsymbol{Y}_{embed}$, the attention matrix $\boldsymbol{A}_{XY} \in \Re^{m\times L}$ is computed, reflecting the attention between data and labels, as shown in Eq. (16).

$$\boldsymbol{X}_{Score} = softmax\left(\boldsymbol{X}_{input}\boldsymbol{W}_X\right) \tag{15}$$

$$\boldsymbol{A}_{XY} = \boldsymbol{X}_{Score}\boldsymbol{Y}_{embed}^{T} \tag{16}$$

During the attention computation for data, inputs undergo an initial linear transformation to be represented in a new vector space, expressed as $\boldsymbol{Q}_X \in \Re^{m\times d_q}$ and $\boldsymbol{V}_X \in \Re^{m\times d_v}$. This facilitates the learning of nonlinear semantic information. Concurrently, the numerical matrix $\boldsymbol{A}_{XY}$, which calculates the joint relevance with labeled data, undergoes a similar transformation and is represented as $\boldsymbol{K}_A \in \Re^{m\times d_k}$,

enabling the computation of joint relevance's semantic representation within this new vector space. In the course of attention computation, the transformed $\boldsymbol{Q}_X$ and $\boldsymbol{K}_A$—corresponding to the relevance matrix—are subject to matrix multiplication to establish attention relations among tokens, influenced by label information. Following normalization via the softmax function, this product is then combined with $\boldsymbol{V}_X$ to yield a new numerical representation of the data, reflecting the attention disparities among tokens. The entirety of this procedure is delineated in Eqs. (17)–(20).

$$\boldsymbol{Q}_X = Linear\left(\boldsymbol{X}_{input}\right) \tag{17}$$

$$\boldsymbol{V}_X = Linear\left(\boldsymbol{X}_{input}\right) \tag{18}$$

$$\boldsymbol{K}_A = Linear\left(\boldsymbol{A}_{XY}\right) \tag{19}$$

$$Attention_{XY}\left(\boldsymbol{X}_{input}\right) = softmax\left(\frac{\boldsymbol{Q}_X \boldsymbol{K}_A^T}{\sqrt{d_k}}\right) V_X \tag{20}$$

Within the model's classification layer, the semantic features discerned by both the diffusion sampling encoder and the label-driven encoder are consolidated and articulated via a linear layer, which facilitates the computation of the predicted outcome, denoted as y_{pred}. Subsequently, the loss function is determined utilizing the cross-entropy technique, as delineated in Eq. (21). In Algorithm 1, we describe the process of the algorithm in the form of pseudocode.

$$Loss = -\sum_i^L y_i \cdot \log\left(y_{pred}^i\right) \tag{21}$$

Algorithm 1: The calculation process of DSLD.

Input: Dataset $S = \{(\boldsymbol{X}_i, y_i)\}_{i=1}^{|N|}$;
Output: Label vector y_{preb} of the test instance;
for $\boldsymbol{i} = \boldsymbol{0} \rightarrow iteration_{num}$ do
// Diffusion Sampling Encoder
Obtain $\boldsymbol{X}_s^i$ according to Eq. (4);
$\boldsymbol{A}^c \leftarrow \sum_i^C$ Score (LinearQ ($\boldsymbol{X}_s^i$), LinearK ($\boldsymbol{X}_s^i$))
$\boldsymbol{A}^{count} \leftarrow Conv2D\ (A^c)$
$Attention_{CC} \leftarrow \boldsymbol{A}^{count} *$ LinearV ($\boldsymbol{X}_i$)
// Label-Driven Encoder
$\boldsymbol{Y}_{embed} \leftarrow Embeding\ (Y)$
$\boldsymbol{A}_{XY} \leftarrow Softmax(\boldsymbol{X}_{input} * \boldsymbol{W}_X) * \boldsymbol{Y}_{embed}^T$
Obtain $\boldsymbol{Q}_X, \boldsymbol{V}_X, \boldsymbol{K}_A$ according to Eqs. (17)–(19);
$Attention_{XY} \leftarrow Attention\_layer(\boldsymbol{Q}_X, \boldsymbol{V}_X, \boldsymbol{K}_A)$
// Classification Layer
$y_{preb} \leftarrow$ Score (LinearQ ($Attention_{CC} + ttention_{XY}$))
end for
Loss $\leftarrow -\sum_i^L \mathrm{y_i} \cdot \log\left(\mathrm{y_{pred}^i}\right)$

4 Experimental Results and Analysis

4.1 Dataset

To assess the effectiveness of the model, this study conducted experiments on seven publicly available datasets, namely, AG News, Yelp Full Review, Yelp Polarity Review, Amazon Full Review, Amazon Polarity Review, DBPedia, and Yahoo! Answers. A detailed description of these datasets is provided in Table 1.

Table 1: Count the number of samples in the training set and test set in each data set, and the length of the longest sample in the data set

Dataset	Classes	Training	Test	Longest length	Task
AG news [25]	4	120 k	7.6 k	135	Topic
Yelp polarity [26]	2	560 k	38 k	1104	Sentiment
Yelp full [26]	5	650 k	50 k	1175	Sentiment
Amazon polarity [27]	2	3600 k	400 k	522	Sentiment
Amazon full [27]	5	3000 k	650 k	520	Sentiment
Yahoo! Answers [26]	10	1400 k	60 k	3998	Topic
DBPedia [28]	14	560 k	70 k	1302	Ontology

4.2 Baseline

In this experiment, a total of five deep learning models that achieved the best results in the task were set as the baseline model.

Transformer [4]: This model features an encoder-decoder architecture and stands out as the pioneering fully attention-based model. Its key strength lies in its efficient parallelization of semantic information computation within textual context.

LBCNN [29]: LBCNN, a label-based convolutional neural network, can capture the importance of individual words in text sequences based on labels. Additionally, it identifies the most influential semantic features within word vectors.

LEAM [30]: This model introduces an attention framework to measure the compatibility between text sequences and labels, thereby facilitating the assessment of embedding compatibility.

WWEM [31]: WWEM incorporates term weighting schemes and n-gram methods, enabling the model to consider both word importance and word order information during the learning of text semantics. This results in the generation of information-rich representations for sentences or documents.

CNLE [12]: CNLE encodes both text and labels into joint representations, fostering interaction between them. This approach enables the model to consider pertinent aspects of both text and labels.

Gated CNNs [32]: The model integrates a gating mechanism into the CNN architecture, which serves to facilitate the efficient transfer of information from preceding layers to the ones that follow.

CWC [33]: The model applies capsule networks to relationship modeling between word embeddings and introduces a novel routing algorithm based on k-means clustering theory to fully explore the relationships among word embeddings.

SLCNN [34]: SLCNN represents documents as three-dimensional tensors within the network, allowing for the comprehensive utilization of positional information in text sentences. This enables the extraction of additional features by analyzing neighboring sentences.

4.3 Experimental Settings

In the model's experimental configuration, we set the learning rate to 0.0001, employed a batch size of 128, and utilized a hidden size of 300. We applied a dropout rate of 0.5 during training, which was conducted for 30 epochs. The dimensions of both word embeddings and label embeddings were fixed at 300. For all convolutional neural networks (CNNs) incorporated into the model, we uniformly set the kernel sizes to 3. Model optimization was carried out using the AdamW optimizer. Additionally, all experiments were performed on an NVIDIA GTX 4090 GPU platform equipped with 24 GB of memory.

4.4 Result and Analysis

Table 2 demonstrates the superior predictive performance of the DSLD model proposed in this paper across all seven benchmark datasets in our experiments. In comparison to the previous optimal model CNLE, the DSLD model exhibited accuracy improvements of 0.16% on the AG News dataset, 0.12% on the Yelp P. dataset, 0.04% on the Yelp F. dataset, and 0.13% on the Amz. P. dataset. Moreover, the DSLD model achieved an accuracy improvement of 0.13% on the Amz. F. dataset. Additionally, the DSLD model outperformed the LEAM model by 0.05% on the Yah. A. dataset and surpassed the LBCNN model by 0.47% on the DBP. dataset. Notably, the DSLD model demonstrated the largest uplift of 0.47% on the Yah. A. dataset, thereby yielding the highest performance improvement among all seven datasets. Across the remaining datasets, the DSLD model consistently displayed accuracy enhancements ranging from 0.1% to 0.5%. These findings collectively establish the proficiency of the DSLD model in accurately classifying text for a variety of text classification tasks, including topic categorization and sentiment analysis.

Table 2: Classification experiment results of each model in seven benchmark datasets

Model	AG news (%)	Yelp P. (%)	Yelp F. (%)	Amz. P. (%)	Amz. F. (%)	Yah. A. (%)	DBP. (%)
Transformer	88.81	96.13	65.34	90.40	54.64	68.89	96.61
LEAM	92.45	95.31	64.09	–	–	77.42	99.02
WWEM	93.20	94.50	61.35	–	–	73.50	98.73
CNLE	94.00	97.13	68.15	96.23	64.18	75.78	99.17
LBCNN	92.90	95.82	64.38	–	–	74.89	99.21
Gated CNNs	90.73	93.75	61.42	–	–	–	98.51
CWC	92.39	96.48	65.85	94.96	60.95	73.85	98.72
SLCNN	91.26	96.01	64.46	93.91	58.11	–	98.76
DSLD (Ours)	**94.16**	**97.25**	**68.19**	**96.36**	**64.23**	**77.89**	**99.33**

As shown in Fig. 4, the DSLD model exhibits the highest accuracy improvement on the Yah. A. dataset. The rationale behind this enhancement lies in the relatively longer average length of sample data in the Yah. A. dataset compared to other datasets. This observation suggests that the model possesses proficiency in capturing semantic information from extended textual contexts.

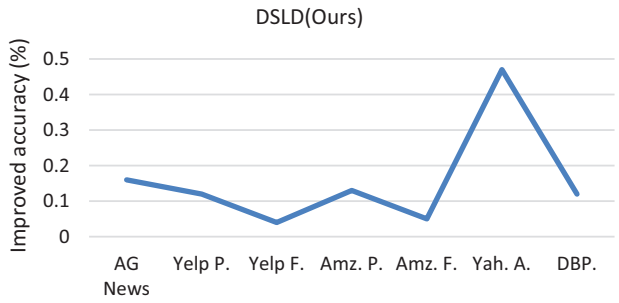

Figure 4: Improvement effect of DSLD model in different benchmark datasets

As shown in Fig. 5a, the DSLD model exhibits the highest improvement on the AG News dataset compared to other models that incorporate label information into text semantic computation. This observation suggests that, in contrast to merely introducing label information into the model's computation process, the diffusion sampling encoder can provide the model with a richer set of polysemous information for learning textual semantic representations. Furthermore, as shown in Fig. 5b, compared with the four models using convolutional methods, the DSLD model achieves an accuracy boost. This phenomenon underscores the capacity of label-driven encoders to compensate for certain latent semantic information that may be absent in the original text.

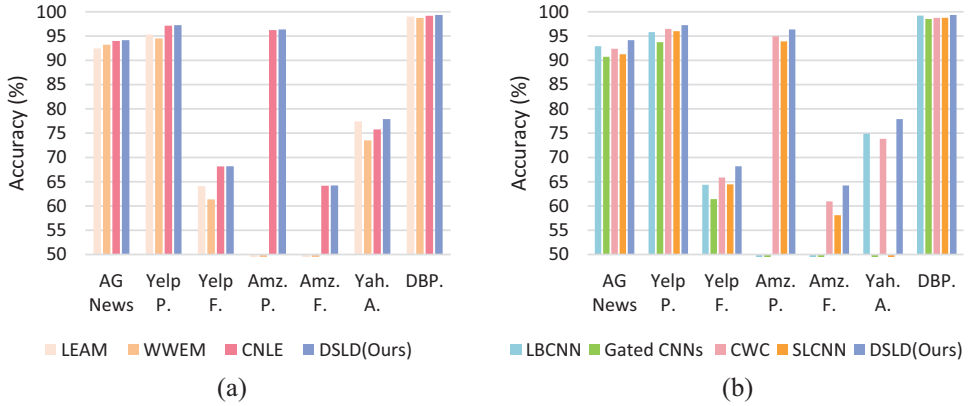

Figure 5: (a) Comparison of accuracy of models using label information in different benchmark datasets, (b) Comparison of the accuracy of the model using convolution method in different benchmark datasets

4.5 Ablation Experiment

To assess the efficacy of individual components within the model and their impact on prediction outcomes, this paper conducted ablation experiments. Ablation experiments entail removing specific model components while preserving the overall model structure. Subsequently, the accuracy, precision, recall, and F1 score of the modified model were compared to those of the DSLD model. A more significant decline in performance indicates the greater importance of the removed component to the DSLD model. Below, we introduce each modified model used in the ablation experiments:

Unlabeled-Driven Encoder: This model exclusively learns text features using the diffusion sampling encoder, serving to evaluate the label-driven encoder's effectiveness.

No Diffusion Sampling Encoder: This model solely relies on the label-driven encoder for text feature learning, allowing an assessment of the diffusion sampling encoder's effectiveness.

Table 3 presents the findings of the ablation study, from which it is discernible that model accuracy is reduced by 1.69% after the removal of the label-driven encoder from the DSLD model. This reduction underscores the essential contribution of label information to the model's semantic processing. A more pronounced decline in accuracy, amounting to 2.45%, ensues upon the excision of the diffusion sampling encoder, highlighting the pivotal role that the diffusion sampling method's semantic information plays in task outcome computations. These variances further reveal that the semantic information derived through label involvement does not fully coincide with that procured from the original data by the diffusion sampling encoder, indicating their potential for synergistic integration.

Table 3: Ablation experiment results

Model	Accuracy (%)	Precision (%)	Recall (%)	F-score (%)
OnlyDSEncoder	92.47	92	92	92
OnlyLDEncoder	91.71	92	92	92
DSLD	**94.16**	**94**	**94**	**94**

4.6 Validation of Model Architecture

In the DSLD model architecture, as depicted in Fig. 1, the diffusion sampling encoder and the label-driven encoder work concurrently to acquire semantic text representations. The output layer integrates features learned by both encoders. To optimize the performance of the diffusion sampling encoder and the label-driven encoder, this study investigates alternative model architectures. In the following sections, we introduce sequential and cross-model architectures and present experimental results in Table 4.

Table 4: Experimental results of model architecture

Model	Accuracy (%)	Precision (%)	Recall (%)	F-score (%)
DSEncoder-LDencoder	92.87	93	93	93
LDencoder-DSEncoder	92.32	92	92	92
Crossover model	92.79	93	93	93
DSLD	**94.16**	**94**	**94**	**94**

In sequential models, the input data undergoes an initial encoding through the first encoder, where the output of the first encoder serves as input for the second encoder. This process facilitates the acquisition of more profound semantic information. The structure of the sequential model, as shown in Fig. 6. In cross-modal models, the label-driven encoder learns label-text-related features that actively participate in the computation process of the diffusion sampling encoder. The cross-model structure is shown in Fig. 7.

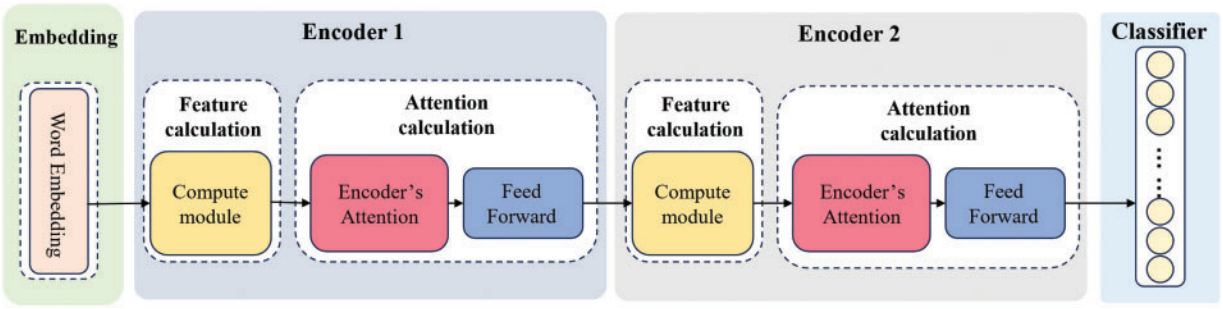

Figure 6: Sequential model architecture diagram

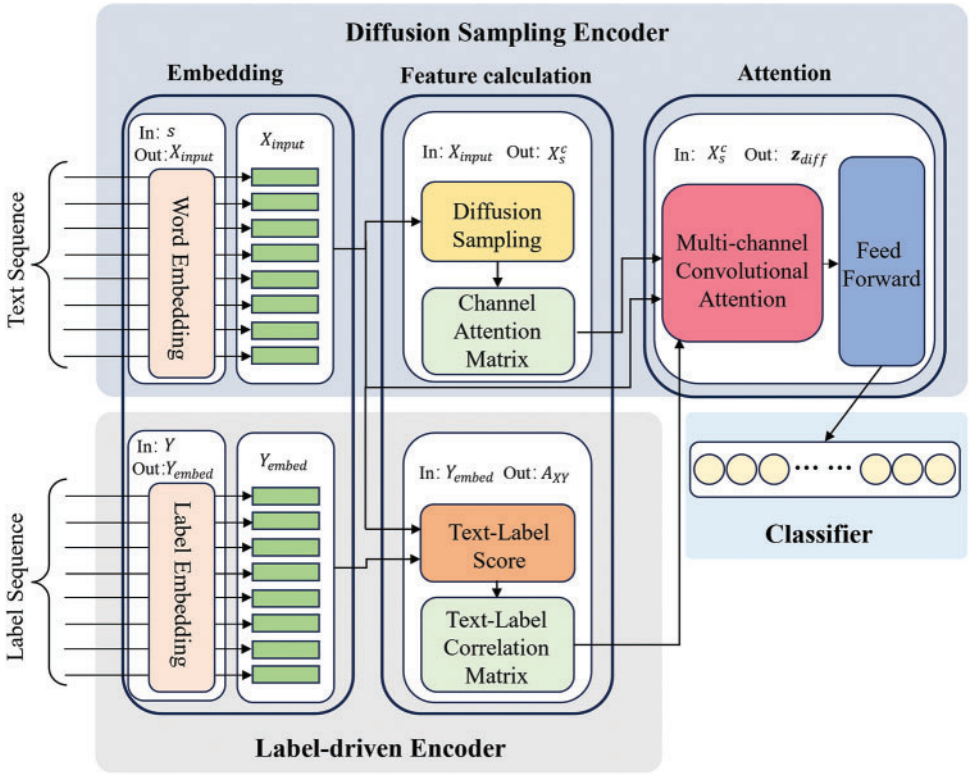

Figure 7: Cross-model architecture diagram

Table 4 demonstrates that the DSLD model achieves the highest model architecture performance. An analysis of this outcome reveals that in the sequential model structure when the diffusion sampling encoder serves as the preceding layer encoder, the semantic relationships learned by the subsequent label-driven encoder between text and labels cannot be effectively applied to compensate for the semantic information encoded by the preceding layer encoder. Conversely, when the diffusion sampling encoder functions as the subsequent layer encoder, it takes as input not the original data but the semantic information represented after processing by the preceding layer encoder. The encoded data, while more complex than the original input data, no longer reflects the same contextual relationships found in the original data. As a result, this affects the model's computational outcomes.

In the cross-model structure, the label-driven encoder learns semantic features that are relatively less diverse compared to the semantic information acquired by the diffusion sampling encoder. Consequently, using the label-driven encoder as compensatory information to guide the diffusion sampling encoder's computations proves less effective than harnessing the richer semantic information obtained through multi-channel sampling by the diffusion sampling encoder. Thus, both the label-driven encoder and the diffusion sampling encoder independently acquire semantic representations of input data, resulting in enhanced feature fusion at the classification layer.

5 Conclusion

In this study, we present the DSLD model which capitalizes on a multi-channel diffusion sampling technique alongside attention mechanisms to discern intricate semantic information in data. The model incorporates the synergistic relevance of labels and data to enhance the precision of semantic representations generated by attentional processes. The DSLD model can capture long-distance semantic dependency information in data while reducing the semantic bias caused by general word vector representation and improving computational accuracy. The efficacy of the DSLD model has been corroborated through trials on benchmark datasets for seven distinct text classification tasks. While current applications of the DSLD model center on text classification, future work will more thoroughly examine the roles of attention methods in semantic analysis. We anticipate extending this model's utility to more comprehensive tasks in sequential data analysis.

Acknowledgement: The authors would like to express their gratitude to Dr. Mengjiao Ma and Dr. Weijian Fan for supervising of this study. The authors would like to express their gratitude for the valuable feedback and suggestions provided by all the anonymous reviewers and the editorial team.

Funding Statement: This research was supported by the Communication University of China (CUC230A013) and the Fundamental Research Funds for the Central Universities.

Author Contributions: Study conception and design: Chunhua Wang, Tong Yi; data collection: Tong Yi; analysis and interpretation of results: Chunhua Wang, Wenqian Shang. Haibin Zhu; draft manuscript preparation: Chunhua Wang, Haibin Zhu. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data that supports the findings of this study are openly available in Torchtext. Torchtext is a Python library for natural language processing that provides the ability to load text data and download corresponding data through the name of the dataset.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] C. Chen, Y. Song, B. Dai, and D. Chen, "Twice attention networks for synthetic speech detection," *Neurocomput.*, vol. 559, pp. 126799, 2023. doi: 10.1016/j.neucom.2023.126799.

[2] Y. Chen, R. Xia, K. Yang, and K. Zou, "DGCA: High resolution image inpainting via DR-GAN and contextual attention," *Multimed. Tools Appl.*, vol. 82, pp. 47751–47771, 2023. doi: 10.1007/s11042-023-15313-0.

[3] J. Zhang, H. Huang, X. Jin, L. D. Kuang, and J. Zhang, "Siamese visual tracking based on critical attention and improved head network," *Multimed. Tools Appl.*, vol. 83, pp. 1589–1615, 2024. doi: 10.1007/s11042-023-15429-3.

[4] A. Vaswani *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 2017, vol. 30, pp. 5998–6008.
[5] Z. Dai, H. Liu, Q. V. Le, and M. X. Tan, "CoAtNet: Marrying convolution and attention for all data sizes," in *Advances in Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 2021, vol. 34, pp. 3965–3977.
[6] L. Zhou, Z. Zhang, L. Zhao, and P. Yang, "Attention-based BiLSTM models for personality recognition from user-generated content," *Inform. Sci.*, vol. 596, pp. 460–471, 2022. doi: 10.1016/j.ins.2022.03.038.
[7] J. Huang, P. Zhao, G. Wang, S. Yang, and J. Lin, "Self-attention-based long temporal sequence modeling method for temporal action detection," *Neurocomput.*, vol. 554, pp. 126617, 2023. doi: 10.1016/j.neucom.2023.126617.
[8] F. Wu, R. Yang, C. Zhang, and L. Zhang, "A deep learning framework combined with word embedding to identify DNA replication origins," *Sci. Rep.*, vol. 11, no. 1, pp. 844, 2021. doi: 10.1038/s41598-020-80670-x.
[9] Y. Tang, "Research on word vector training method based on improved skip-gram algorithm," *Adv. Multimedia*, vol. 2022, pp. 4414207, 2022. doi: 10.1155/2022/4414207.
[10] G. Wang *et al.*, "Joint embedding of words and labels for text classification," in *Proc. 56th Annu. Meeting of the Assoc. for Computat. Linguist.*, 2018, pp. 2321–2331.
[11] H. Liu, G. Chen, P. Li, P. Zhao, and X. Wu, "Multi-label text classification via joint learning from label embedding and label correlation," *Neurocomput*, vol. 460, pp. 385–398, 2021. doi: 10.1016/j.neucom.2021.07.031.
[12] M. Liu, L. Liu, J. Cao, and Q. Du, "Co-attention network with label embedding for text classification," *Neurocomput.*, vol. 471, pp. 61–69, 2022. doi: 10.1016/j.neucom.2021.10.099.
[13] C. Beghtol, "Bibliographic classification theory and text linguistics: Aboutness analysis, intertextuality and the cognitive act of classifying documents," *J. Doc.*, vol. 42, no. 2, pp. 84–113, 1986. doi: 10.1108/eb026788.
[14] L. C. Cheng, Y. L. Chen, and Y. Y. Liao, "Aspect-based sentiment analysis with component focusing multi-head co-attention networks," *Neurocomput.*, vol. 489, pp. 9–17, 2022. doi: 10.1016/j.neucom.2022.03.027.
[15] Y. Liu, P. Li, and X. Hu, "Combining context-relevant features with multi-stage attention network for short text classification," *Comput. Speech Lang.*, vol. 71, pp. 101268, 2022. doi: 10.1016/j.csl.2021.101268.
[16] Y. Chen, R. Xia, K. Yang, and K. Zou, "DARGS: Image inpainting algorithm via deep attention residuals group and semantics," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 35, no. 6, pp. 101567, 2023. doi: 10.1016/j.jksuci.2023.101567.
[17] Y. Wang, W. Zheng, Y. Cheng, and D. Zhao, "Two-level label recovery-based label embedding for multi-label classification with missing labels," *Appl. Soft Comput.*, vol. 99, pp. 106868, 2021. doi: 10.1016/j.asoc.2020.106868.
[18] T. Ni, Y. Ding, J. Xue, K. Xia, X. Gu and Y. Jiang, "Local constraint and label embedding multi-layer dictionary learning for sperm head classification," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 17, no. 3s, pp. 1–16, 2021. doi: 10.1145/3458927.
[19] W. Liu *et al.*, "Volumetric segmentation of white matter tracts with label embedding," *Neuroimage*, vol. 250, pp. 118934, 2022. doi: 10.1016/j.neuroimage.2022.118934.
[20] J. Tang, M. Qu, and Q. Mei, "PTE: Predictive text embedding through large-scale heterogeneous text networks," in *Proc. KDD*, 2015, pp. 1165–1174.
[21] H. Zhang, L. Xiao, W. Chen, Y. Wang, and Y. Jin, "Multi-task label embedding for text classification," in *Proc. 2018 Conf. Empirical Methods in Natural Lang. Process.*, 2018, pp. 4545–4553.
[22] N. Pappas and J. Henderson, "GILE: A generalized input-label embedding for text classification," *Trans. Assoc. Computat. Linguist.*, vol. 7, pp. 139–155, 2019. doi: 10.1162/tacl_a_00259.
[23] N. Liu, Q. Wang, and J. Ren, "Label-embedding bi-directional attentive model for multi-label text classification," *Neural Process. Lett.*, vol. 53, pp. 375–389, 2021. doi: 10.1007/s11063-020-10411-8.
[24] M. Abdel Aal, S. Djennadi, O. Abu Arqub, and H. Alsulami, "On the recovery of a conformable time-dependent inverse coefficient problem for diffusion equation of periodic constraints type and integral overposed data," *Math. Probl. Eng.*, vol. 2022, pp. 5104725, 2022. doi: 10.1155/2022/5104725.

[25] J. Wang, Z. Wang, D. Zhang, and J. Yan, "Combining knowledge with deep convolutional neural networks for short text classification," in *Proc. IJCAI*, 2017, vol. 350, pp. 3172077–3172295.
[26] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 2015, vol. 1, pp. 649–657.
[27] J. McAuley and J. Leskovec, "Hidden factors and hidden topics: Understanding rating dimensions with review text," in *Proc. 7th ACM Conf. on Recommender Syst.*, 2013, pp. 165–172.
[28] J. Lehmann *et al.*, "DBpedia–a large-scale, multilingual knowledge base extracted from Wikipedia," *Semant. Web.*, vol. 6, no. 2, pp. 167–195, 2015. doi: 10.3233/SW-140134.
[29] C. Wang and C. Tan, "Label-based convolutional neural network for text classification," in *Proc. 5th Int. Conf. on Control Eng. Artificial Intell.*, 2021, pp. 136–140.
[30] C. Du, Z. Chen, F. Feng, L. Zhu, T. Gan and L. Nie, "Explicit interaction model towards text classification," in *Proc. AAAI Conf. Artificial Intell.*, vol. 33, no. 1, pp. 6359–6366, 2019. doi: 10.1609/aaai.v33i01.33016359.
[31] H. Ren, Z. Q. Zeng, Y. Cai, Q. Du, Q. Li and H. Xie, "A weighted word embedding model for text classification," in *Database Syst. Adv. Appl.: 24th Int. Conf.*, Chiang Mai, Thailand, Springer International Publishing, 2019, vol. 24, pp. 419–434.
[32] J. Sun, R. Jin, X. Ma, J. Park, K. Sohn and T. Chung, "Gated convolutional neural networks for text classification," in *Adv. Comput. Sci. Ubiquitous Comput.: CSA-CUTE 2019*, Singapore, Springer, 2021, pp. 309–316.
[33] H. Ren and H. Lu, "Compositional coding capsule network with k-means routing for text classification," *Pattern Recogn. Lett.*, vol. 160, pp. 1–8, 2022. doi: 10.1016/j.patrec.2022.05.028.
[34] A. Jarrahi, R. Mousa, and L. Safari, "SLCNN: Sentence-level convolutional neural network for text classification," arXiv preprint arXiv:2301.11696, 2023.