DPDANet: An Improved DPCNN Text Classification Model Integrating Dense Connections and Self-Attention Mechanisms

Authors: Luo Huayu, Wang Dongmei, Zhang Yinghui, Lin Wei, Chen Chen, Wang Dongmei

Date: 2025-05-05T00:00:00+00:00

Abstract

[Objective] To address the need for efficient sentiment analysis of massive comment data, we propose the DPDANet model to enhance text classification performance. [Method] DPDANet, built upon BERT, integrates dense connections and attention mechanisms by optimizing the inter-layer connection strategy of DPCNN, which enhances feature flow and information reuse capabilities, thereby more efficiently utilizing shallow features and effectively reducing computational complexity. [Results] We conducted comparative experiments between DPDANet and eight models including BERT-based TextCNN, CNN-LSTM, DPCNN, DPCNN-BiGRU, Transformer, XLSTM, BERT, and DPDB-Net. On four text classification datasets, DPDANet achieved excellent accuracy scores of 0.6679, 0.9307, 0.9278, and 0.6242, respectively, representing improvements of 6.47%, 1.32%, 0.72%, and 3.52% over DPCNN. [Limitations] The model still exhibits insufficient generalization capability in scenarios with extremely short texts and multi-class imbalance. [Conclusion] DPDANet demonstrates superior performance and efficiency across numerous text classification tasks and possesses promising application prospects.

Full Text

Preamble

DPDANet: An Improved DPCNN Model for Text Classification with Dense Connections and Self-Attention Mechanism

Luo Huayu[1], Wang Dongmei[1], Zhang Yinghui[1], Lin Wei[1], Chen Chen[1]
[1](School of Data Science and Engineering, South China Normal University, Guangdong 516622, China)

Abstract:
[Objective] In response to the demand for efficient sentiment analysis of large-scale review data, this study proposes the DPDANet model to enhance text classification performance. [Methods] The BERT-based DPDANet incorporates dense connections and an attention mechanism. By refining the inter-layer connection strategy of the DPCNN architecture, it enhances feature propagation and information reuse, thereby facilitating more efficient exploitation of shallow features while effectively reducing computational complexity. [Results] Comparative experiments were conducted between DPDANet and eight BERT-based models, including TextCNN, CNN-LSTM, DPCNN, DPCNN-BiGRU, Transformer, XLSTM, BERT, and DPDBNet. On four text classification datasets, DPDANet achieved outstanding accuracy scores of 0.6679, 0.9307, 0.9278 and 0.6242, representing improvements of 6.47%, 1.32%, 0.72% and 3.52%, respectively, over the baseline DPCNN model. [Limitations] The model still exhibits limited generalization capability in scenarios involving extremely short texts and imbalanced multi-class distributions. [Conclusions] DPDANet demonstrates superior performance and efficiency across a variety of text classification tasks, indicating strong potential for practical application.

Keywords: DPDANet; Text Classification; DPCNN; Self-Attention Mechanism

Introduction

Text classification stands as one of the core tasks in Natural Language Processing (NLP). In the digital era, the rapid proliferation of the internet and the explosive growth of online information have led to an unprecedented surge in text data volume. Various forms of textual data, including social media posts, online news, emails, and user reviews, have substantially increased the complexity and challenges in information extraction and decision-making processes [1]. Against this backdrop, text classification has emerged as an efficient and automated text processing tool with widespread applications across numerous domains.

For instance, text classification techniques can assist physicians in making more precise diagnostic judgments based on patient medical records [2], while automatic categorization of news texts helps improve the efficiency and accuracy of personalized recommendation systems [3]. Sentiment analysis, as a crucial application branch of text classification, demonstrates broad development prospects [4][5]. This technique aims to meticulously identify and classify emotions, attitudes, or stances embedded in textual data, enabling effective recognition and categorization of public sentiment. It has been widely applied to data processing tasks on online platforms such as social media, user reviews, and product evaluations, assisting government agencies and enterprises in perceiving public opinion dynamics in real time and making timely, informed decisions for precise monitoring and effective guidance of public sentiment.

In the context of the new era and big data, influenced by factors such as in-

ternet slang, online comments generally exhibit linguistic diversity and complexity. Traditional categories of seven psychological emotions can no longer comprehensively capture the sentiment characteristics of online comments in modern contexts. Therefore, building upon the conventional emotion classification framework, this paper further introduces three special emotion types that frequently appear in online contexts—humor, sarcasm, and confusion—to more accurately and appropriately reflect the actual distribution of emotions in internet corpora.

In NLP, sentiment classification methods have already established a solid foundation. The rise of neural networks has gradually replaced traditional machine learning approaches, becoming the mainstream methodology for NLP research both domestically and internationally. In recent years, NLP technologies based on deep networks have achieved remarkable progress, particularly through deep architectures constructed by stacking convolutional layers, which have yielded exceptional results in NLP tasks [6][7]. Deep network architectures possess stronger feature representation capabilities, capturing complex semantic information and finer-grained semantic features in text through hierarchical abstract learning and multi-level semantic feature modeling.

However, deep networks are typically accompanied by high training costs and a high incidence of gradient explosion or vanishing. Therefore, how to ensure efficient information flow in deep networks while reducing their complexity and further improving text classification performance remains a critical research challenge. This paper proposes an improved Deep Pyramid Convolutional Neural Network [8] (DPCNN)—the Deep Pyramid Dense Attention Network (DPDANet). This model significantly enhances the reuse of low-level feature information while optimizing information flow in deep networks by introducing dense connections and self-attention mechanisms. This design effectively reduces the number of model parameters and improves the model' s ability to capture relationships between words.

Related Work

In recent years, deep learning has made significant advances in the NLP domain, with research continuously evolving from traditional shallow network models to more complex deep network architectures. These models consistently improve text classification performance by extracting and optimizing feature representations. This section explores various deep learning methods and deep network structures proposed for text classification tasks.

2.1 Deep Learning Models

Regarding word vector representations, the field has witnessed an evolution from static models (such as Word2Vec [9] and GloVe [10]) to context-based dynamic models (such as ELMo [11], BERT [12], GPT [13], and XLNet [14]). Compared to static models, the latter can more accurately capture how word meanings

change with context, significantly enhancing word representation capabilities and downstream task performance.

Among these, BERT demonstrates exceptional generalization ability and robustness in tasks such as sentiment analysis, particularly achieving excellent results through fine-tuning on small-scale datasets. For example, scholars Sun Dandan et al. [15] employed BERT as an embedding layer to enhance model performance in their research on network public opinion analysis during the COVID-19 pandemic; Zhang Jing et al. [16] also utilized BERT in their intelligent classification of police report texts using BERT-DPCNN. In contrast, although models like GPT and XLNet possess stronger modeling capabilities on large-scale corpora, they often face issues such as poor adaptability, difficult fine-tuning, and slower inference speeds in few-shot scenarios. Therefore, considering both the expressive power of word vector representations and practical application requirements, this paper selects BERT as the embedding layer to balance performance and efficiency.

In the deep learning domain, one of the earliest models applied to text classification is the Text Convolutional Neural Network (TextCNN), which Kim et al. [17] applied to sentiment analysis tasks. However, since CNNs cannot handle long-distance dependencies, Elman [18] proposed the Recurrent Neural Network (RNN), whose temporal dependency characteristics offer clear advantages in capturing global contextual information. Nevertheless, RNNs suffer from gradient explosion and difficulty in modeling long-term dependencies. To address these issues, Sepp Hochreiter [19] proposed the Long Short-Term Memory (LSTM) network. Subsequent variants such as Bidirectional Long Short-Term Memory [20] (BiLSTM) and Gated Recurrent Unit [21] (GRU) have demonstrated significant improvements in capturing global information and reducing computational complexity.

Vaswani et al. [22] noted that traditional attention mechanisms overly rely on external information and cannot effectively model internal relationships within sequential data. To address this, they introduced a self-attention mechanism architecture in the encoder to learn text representations. Consequently, Vaswani et al. abandoned traditional convolutional and recurrent structures and proposed the Transformer model. Although Transformer performs exceptionally well on large-scale datasets, its stacked self-attention architecture incurs high computational complexity, placing it at a certain disadvantage in small-scale sentiment analysis. On one hand, the computational cost of self-attention mechanisms grows quadratically, consuming substantial resources; on the other hand, Transformer neglects the expression of local sentiment features when modeling global context, causing the model to exhibit insufficient stability and poor robustness in sentiment analysis scenarios with limited sample sizes.

2.2 Deep Network Architectures

Deep network architectures refer to neural network structures that extract complex features by stacking multiple layers. These architectures enhance model performance across various tasks by increasing network depth, enabling them to capture multi-level information from data. In computer vision, strategies of deepening network layers have already achieved remarkable results.

The Deep Pyramid Convolutional Neural Network (DPCNN), proposed by Johnson' s team, holds broad prospects in practical applications. This model effectively addresses the problem that shallow convolutions struggle to capture long-distance dependencies in text. Moreover, by fixing the number of feature maps during downsampling, it substantially reduces the computational complexity of deep network structures. Since its proposal, DPCNN has been widely adopted. For instance, Zhao et al. [23] constructed a system for addressing online malicious language issues based on the multilingual model XLM-Roberta and DPCNN, validating its effectiveness on official test datasets. Yang et al. [24] proposed an intelligent legal document recommendation method combining DPCNN with capsule networks, solving the problem of low accuracy in legal document recommendation caused by imbalanced category distribution and significantly improving model performance. Xian et al. [25] proposed a short text classification algorithm combining DeBERTa and DPCNN to address the sparsity, real-time requirements, and irregularity of news headline data, with their model achieving superior accuracy compared to other models. Yu et al. [26] combined sequence modeling with temporal convolutional networks and DPCNN to propose the Deep Pyramid Temporal Convolutional Network, significantly improving accuracy in paragraph-level text classification. Beck et al. [27] proposed the Extended Long Short-Term Memory (XLSTM) network based on LSTM, which significantly enhances model performance and scalability—particularly evident in complex tasks—by stacking LSTM networks and introducing exponential gating mechanisms and novel memory structures. However, it performs poorly on small-scale multi-class sentiment classification tasks and still requires high computational costs, which is further validated in subsequent experiments in this paper.

The proposal of DenseNet [28] provides DPCNN with a new inter-layer connection approach—dense connections—which enhances feature reuse capability without increasing parameter count. Building upon this, this paper constructs DPDANet by integrating dense connections and attention mechanisms into DPCNN. This approach reuses low-level information while ensuring sufficient modeling of word relationships. The improved DPCNN model demonstrates superior performance in both parameter scale and accuracy.

DPDANet Text Classification Model

This section first outlines the overall workflow of DPDANet, analyzes its key improvements over DPCNN, and then elaborates on the proposed DPDANet

architecture and its core design philosophy.

3.1 Overall Model Architecture

The structure of DPDANet is illustrated in Figure 1 [Figure 1: see original paper]. The model' s first layer utilizes the hidden state from BERT' s final layer as the text' s word vector representation, providing a rich semantic foundation for subsequent feature extraction. DPDANet first passes through a bottleneck layer with a self-attention mechanism to integrate information from BERT and preliminarily extract semantic information. Incorporating self-attention early in the network helps capture global contextual information at the initial stage of feature extraction, mitigating semantic loss caused by convolutions. Subsequently, multiple convolutional blocks are stacked, each comprising a 1/2 max-pooling layer and two convolutional layers. The first convolutional layer serves as a transition layer to compress features from all previous layers, where the compression factor is bn; the second convolutional layer learns more refined features on extremely small feature maps and feeds them into the global state. Afterward, a self-attention mechanism is introduced, enabling the model to better focus on more important features under smaller feature maps. This structural component is named the Dense Attention Block (DAB). The proposed model learns sentence semantic representations through ten convolutional layers, with the following main characteristics: (1) In each DAB, features are first extracted through convolutional layers with a small number of channels. During each iteration, the extracted features are concatenated with features obtained from all previous layers. The dense connection approach achieves feature reuse and ensures maximum information flow between layers. Meanwhile, DPDANet inherits DPCNN' s strategy of halving computational cost through a 1/2 pooling layer after every two convolutions, thereby significantly reducing computational complexity. (2) The model introduces a self-attention mechanism within DABs, enabling it to adaptively focus on key information, enhance modeling of dependencies between words, and thus more effectively capture textual semantic information.

3.2 Encoding Layer

In the information age, language often exhibits complex polysemy. To fully utilize word-level semantic information from input text, DPDANet employs BERT as the encoding layer. Let the input sentence be $(x_1, x_2, \ldots, x_n)$, where x_i represents the i-th character of the sentence and n is the sentence length. BERT adds special tokens $x_{[\mathrm{CLS}]}$ and $x_{[\mathrm{SEP}]}$ at the beginning and end of the sentence, respectively, then converts the sentence into an embedding sequence $(E_0, E_1, E_2, \ldots, E_n)$ through the embedding layer, where each $E_i \in \mathbb{R}^d$ and d is the hidden state dimension. Each E_i is obtained by summing the following three components, with its mathematical representation as:

Subsequently, BERT feeds this embedding sequence into a network structure stacked with m Transformer modules, where each layer models dependencies

within the sentence through attention mechanisms. This stacking process can be recursively represented as:

DPDANet selects the hidden state from BERT' s final layer as the input representation for the downstream classification model.

3.3 Dense Attention Block

Convolution Operation: The core of the dense attention block consists of two convolutional layers. Convolution operations extract features by performing local weighted summation on input data through sliding convolutional kernels until the entire input is covered, ultimately generating feature representations. This process can be expressed as:

$$Y_{ij} = \sigma(W_j \cdot x_{i:i+h-1} + b_j)$$

where Y_{ij} represents the output of the j-th convolutional kernel applied to the input region $x_{i:i+h-1}$, h denotes the convolutional kernel size, W_j is the weight matrix of the j-th convolutional kernel, b_j is the bias term, and σ is the ReLU activation function with the following mathematical expression:

$$\sigma(x) = \max(0, x)$$

In DPCNN, maintaining feature map dimensions is achieved by using convolutions with kernel size 3, stride 1, and zero padding at both ends. This operation not only effectively expands the model' s receptive field but also fully captures local features of sentences, ensuring emotional information is adequately expressed in deep networks. Moreover, this high-fidelity local feature extraction provides more discriminative feature map inputs for subsequent self-attention mechanisms, helping the model more accurately model key semantic relationships in text.

Dense Connections: Since equal-length convolutions are employed in convolutional layers without changing input sequence length, this type of convolution provides a structural foundation for concatenating network layer outputs along the channel dimension. Specifically, each convolutional layer in DAB generates only k features, where k is an adjustable hyperparameter called the growth rate. In the dense connection structure, each transition layer receives feature maps from all previous layers as its input. The mathematical expression is as follows:

$$x_l = H_l([x_0, x_1, \ldots, x_{l-1}])$$

This connection approach achieves maximum feature reuse, enabling the network to efficiently and rationally utilize shallow network features even in deep architectures.

Self-Attention Mechanism: To achieve more precise information flow modeling, DPDANet introduces self-attention mechanisms after the initial bottleneck layer and after every two convolutional layers in DAB. This mechanism can effectively weight features extracted by convolutional layers, capture dependencies between features, and enable the network to focus on more critical information.

Specifically, the input sequence generates query vectors (Q), key vectors (K), and value vectors (V) through different linear transformations. Each attention head calculates dependency degrees between elements within the sequence to generate attention weights, which are then used to weight the input through weighted summation to obtain representations for each element. The mathematical operations involved in this mechanism are shown below:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

where d_k represents the dimension of key vectors to ensure stable model training.

Pyramid Structure: The pyramid structure refers to downsampling operations using a max-pooling layer with size 3 and stride 2. The primary function of this pooling layer is to retain key information extracted by convolutional layers while halving the sequence length. Specifically, the pooling layer generates new representations of text within a region by extracting the maximum value from three consecutive vector components, but operates with a stride of 2 (i.e., skipping one element between every three elements).

3.4 Output Layer

After passing through DPDANet' s recurrent structure, feature maps are compressed along the spatial dimension through a global max-pooling layer to extract global feature information. These compressed features are then fed into a fully connected layer, which maps them to corresponding classification labels to accomplish the final classification task.

3.5 Loss Function

DPDANet is applied to multi-classification tasks for novel emotional expressions emerging in online comments within the context of the new era. Therefore, the model adopts the cross-entropy loss function, which guides the model to better fit the data in classification tasks by minimizing the difference between the predicted probability distribution and the true labels. Its mathematical expression is as follows:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(\hat{y}_i)$$

where N represents the total number of samples, y_i is the true label of the i-th sample, and $\hat{y}_i$ is the predicted probability value for the i-th sample.

Experiments and Analysis

This section designs and conducts a series of experiments to comprehensively validate the effectiveness and robustness of the proposed model. These experiments cover multiple benchmark datasets and compare against state-of-the-art models.

4.1 Experimental Setup

Experiments were conducted on a 64-bit Windows 11 operating system. The experimental environment is equipped with an AMD Ryzen 7 5800H with Radeon Graphics processor and an NVIDIA GeForce RTX 3060 Laptop GPU. Model construction is based on Python 3.9 and CUDA 12.3, within PyCharm 2023.

4.2 Experimental Design

Datasets: This study selects four real-world datasets to evaluate the proposed DPDANet model, where AVG_{length} represents the average sentence length in the corpus. The first dataset was obtained through web scraping from Douyin (TikTok). Sentences in this dataset commonly contain extensive internet slang, complex semantics, and noisy information, making it highly suitable for testing the model's noise resistance. Considering the requirements of sentiment analysis tasks, we introduced three emotion types common in modern online comments —humor, sarcasm, and confusion—to this dataset. For data annotation, we employed manual labeling to annotate the obtained text data with emotion categories. Ultimately, this task was divided into 11 emotion categories, as shown in Table 1 :

Table 1 Eleven-Class Emotion Classification

The latter three datasets are used to more broadly validate the model' s generalization capability: AGNews [29], a well-known dataset widely used for topic classification research; and the third and fourth datasets are the sentiment binary-classification dataset SST-2 [30] and the News Category Dataset from Kaggle [31].

Table 2 Datasets Information

This study conducted statistical analysis on sentence length distribution across all datasets, as shown in Figure 2 [Figure 2: see original paper]. The results show that the selected datasets exhibit good diversity in terms of sentence length. Specifically, AGNews and Kaggle datasets primarily contain medium to long sentences, while SST and TikTok datasets consist mainly of short sentences. This diverse distribution helps comprehensively evaluate the model' s generalization capability and performance across different text lengths.

Evaluation Metrics: To evaluate text classification performance, this paper adopts the widely used metric—Accuracy (ACC):

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}$$

where TP represents the number of samples correctly predicted as positive, TN represents the number of samples correctly predicted as negative, FP represents the number of samples incorrectly predicted as positive, and FN represents the number of samples incorrectly predicted as negative.

Additionally, to compare the functionality of different components, this paper employs Relative Gain (RG) to assess the impact of each component on model performance improvement. The metric is calculated as follows:

$$\mathrm{RG} = \frac{\mathrm{ACC}_{\text{unit}} - \mathrm{ACC}_{\text{non-unit}}}{\mathrm{ACC}_{\text{non-unit}}} \times 100\%$$

where $\mathrm{ACC}_{\text{unit}}$ represents the accuracy of the model with a certain component added, and $\mathrm{ACC}_{\text{non-unit}}$ represents the accuracy of the model with that component removed.

Hyperparameters and Network Design: To ensure experimental validity, all model parameters are kept consistent. Specific hyperparameter designs and network structures for each layer are shown in the table.

Table 3 Hyperparameter Settings

Training Details: DPDANet is built in the PyTorch environment, with network weights initialized using the Kaiming method. The Adam optimizer is employed during model training to adaptively adjust the learning rate. A learning rate decay factor is also introduced to accelerate model convergence. Regarding training strategy, early stopping is adopted to terminate training prematurely based on validation set performance monitoring. To mitigate overfitting risks and reduce computational costs, this paper only fine-tunes BERT' s embedding layer and its last three layers.

4.3 Performance Comparison

This section addresses a core question: Can the proposed DPDANet effectively improve character-level text classification accuracy? First, this paper compares DPDANet with various baseline models, including classic shallow network models and some deep learning models. Additionally, to verify whether the fusion of BERT and self-attention mechanisms significantly enhances model performance, a variant of DPDANet—Deep Pyramid Dense Block Network (DPDBNet)—is designed.

Comparative Analysis of Different Methods: Under identical benchmark conditions, this section conducts comparative experiments between DPDANet and eight other models, all using BERT outputs as the word embedding layer to ensure consistency and comparability. The first two models are shallow neural networks, while the rest are deep neural networks. The selected comparison models are as follows:

1. **TextCNN:** A shallow Convolutional Neural Network (CNN) that takes word vectors generated by BERT as input and extracts different n-gram features through three parallel convolutional layers with kernel sizes of 3, 4, and 5.
2. **C-LSTM:** Introduces an LSTM structure on top of TextCNN. Convolutional layers extract local features, while LSTM captures long-range dependencies in text through its recurrent structure. This paper uses an LSTM with 128 hidden units to model long-distance dependencies in text.
3. **DPCNN:** A Deep Pyramid Convolutional Neural Network that captures deep-level text features through multiple convolutional layers. It employs a pyramid structure to progressively downsample convolutional layer outputs, thereby enhancing the model' s feature representation capability.
4. **DPCNN-BiGRU [32]:** A variant that introduces Bidirectional Gated Recurrent Unit (BiGRU) on top of DPCNN to jointly extract local features and global semantic information. BiGRU captures contextual relationships in text, while DPCNN strengthens long-distance feature modeling through its deep pyramid structure, with GRU hidden layer neurons set to 128.
5. **xLSTM:** The xLSTM model introduces exponential gating mechanisms to flexibly control information flow, thereby more effectively capturing long-distance dependencies and mitigating gradient vanishing issues in long-sequence training. Additionally, innovative memory structures further optimize information storage and transmission efficiency, enhancing modeling capabilities for complex text patterns, with hidden layer neurons also set to 128.
6. **Transformer:** Composed of encoders and decoders, each containing multiple stacked self-attention layers.
7. **BERT:** Directly feeds the generated CLS representation into a fully connected layer for classification, leveraging its powerful contextual understanding capability.
8. **DPDBNet:** A model variant without multi-head attention mechanisms to verify the specific contribution of attention mechanisms to model performance.

Table 4 Classification results of DPDANet and other baseline models on different datasets

Preliminary Experimental Analysis: As shown in Table 4, DPDANet demonstrates excellent performance on Douyin, AGNews, and Kaggle datasets, achieving accuracies of 0.6679, 0.9307, and 0.6242, respectively, surpassing main-

stream models including Transformer. This highlights its superior performance in both topic and sentiment multi-classification tasks. Meanwhile, DPDANet' s outstanding performance on the eleven-class dataset constructed from Douyin underscores its unique advantages in processing social media data containing internet slang and high noise, demonstrating stronger classification capability and noise robustness. On the SST-2 sentiment binary-classification dataset, although DPDANet is slightly inferior to XLSTM, it still achieves relatively good results. XLSTM' s excellent performance on this dataset further proves the significant potential of LSTM-based networks in capturing sentiment-related features. However, for small-scale multi-class sentiment analysis tasks, the DPDANet proposed in this paper performs more excellently.

4.4 Ablation Studies

Analysis of Different Components: Based on the accuracies of each model in the table above, to further analyze the improvements brought by dense connections and attention mechanisms, this paper presents Table 5 to examine the impact of different components on experimental results:

Table 5 Relative Gains of Different Components

In the table, model gain represents the improvement of DPDANet over DPCNN, dense block gain reflects the performance improvement of DPCNN after introducing dense connections, and attention gain indicates the relative gain brought by introducing self-attention mechanisms.

The analysis results in Table 5 show that introducing dense connections brings performance improvements across all datasets. This may be because such a structure promotes efficient inter-layer information propagation, helping the model better learn and capture latent text features. Additionally, attention gain results demonstrate that introducing self-attention mechanisms also effectively enhances model performance, as this mechanism strengthens the model' s ability to focus on important words.

Optimal Growth Rate: To investigate the impact of different growth rates on model performance, this paper evaluates DPDANet's performance under various growth rate settings on the SST-2 dataset using error rate as the evaluation metric, while simultaneously recording training time to further describe the impact of growth rate on training efficiency. Experimental results are shown in Figure 3 [Figure 3: see original paper].

Based on the results, the following observations can be made: As the growth rate increases from 4 to 32, model performance shows a gradual improvement trend, while training time also increases significantly. Notably, when the growth rate is set to 32, it achieves optimal performance in all comparisons, while training efficiency remains at a relatively high level, achieving a good balance between performance and efficiency. When the growth rate reaches 16, training time begins to increase rapidly; when the growth rate further increases to 64, model

performance instead begins to decline, failing to continue improving.

This phenomenon can be explained as follows: An excessively small growth rate leads to DPDANet obtaining fewer features during the dense connection process, lacking effective understanding of deep text semantics. This results in information integrated into the global state that may be relatively one-sided, making it difficult to effectively capture details and contextual relationships in text, thereby affecting the model' s semantic understanding and contextual adaptability.

On the other hand, an excessively large growth rate may cause feature redundancy, weakening the model's generalization ability when processing fine-grained semantics. An overly high growth rate not only increases computational complexity but may also lead to excessive fusion of features between different layers, making it difficult for the model to focus on key information and affecting both model performance stability and training efficiency.

Computational Complexity: As mentioned earlier, the introduction of dense blocks significantly enhances feature transmission capability while substantially reducing the number of DPCNN parameters. This section focuses on comparing parameter scale differences between DPDANet and DPCNN at the same network depth. To more intuitively present the impact of dense connections on model structure, Table 6 lists the number of convolutional layer parameters for three models—DPDBNet, DPDANet, and DPCNN—at different depths.

Table 6 Model Parameters at Different Depths

The table shows that compared to DPCNN at the same network depth, DPDANet' s dense connection approach significantly reduces the number of parameters. Furthermore, as network depth increases, the parameter growth rate of the proposed model is much lower than that of DPCNN.

Figures 4 [Figure 4: see original paper] and 5 [Figure 5: see original paper] present the performance of DPCNN and DPDANet at different depths on the binary-classification SST dataset.

Figure 4 Performance of DPDANet at different growth rates
Figure 5 Performance Comparison of DPCNN at Different Depths

Based on observations, the following conclusions can be drawn: First, as network depth increases, both DPCNN and DPDANet show declining error rates. However, after reaching a certain depth, performance improvements plateau with minor fluctuations. Specifically, when both DPCNN and DPDANet contain 10 convolutional layers, they achieve their lowest error rates, significantly outperforming models with other depths.

Second, under all depth conditions, DPDANet consistently outperforms DPCNN, particularly in training speed and final performance, demonstrating clear advantages. DPDANet' s superiority may be attributed to the introduction of dense connections, which enhance information flow efficiency and

prevent gradual feature disappearance in deep networks. As DPDANet' s depth increases, model performance continues to improve. However, when depth reaches 10 layers, performance begins to saturate, and further increasing depth (e.g., to 12 layers) leads to a slight increase in error rate, possibly due to information loss caused by overly deep networks.

4.5 Feature Reuse

DenseNet has already proven that dense connections can enhance a model' s feature reuse capability. This section primarily examines whether introducing dense blocks in the proposed model can also achieve feature reuse. Experiments were conducted on a trained DPDANet_{10} model. By calculating the average absolute filter weights of inter-layer connections, we demonstrate the dependency relationship of convolutional layers on all previous layers. Among the 10 convolutional layers, transition layers in dense blocks exhibit characteristics of dense connections. Therefore, this paper selected five transition layers for heatmap visualization. As shown in the figure, darker colors at position (l, s) indicate stronger dependency of layer l on features generated by layer s.

Figure 6 [Figure 6: see original paper] Performance Comparison of DPCNN at Different Depths

In the heatmap matrix, darker colors in the bottom-right corner, such as at position (5,5), indicate that layer 5 strongly depends on features generated by all previous layers. This phenomenon demonstrates the effectiveness of introducing dense connections to DPCNN, as dense connections enable deep convolutional layers to fully utilize all shallow-layer information, thereby enhancing feature reuse capability.

4.6 Discussion

The first comparative experiment in Section 4.3 demonstrates that introducing dense attention blocks into the DPCNN model brings a certain degree of accuracy improvement on both public datasets and the Douyin dataset. DPDANet's performance improved by 1.32%, 0.72%, 3.52%, and 6.47% over DPCNN across the four datasets. This indicates that introducing dense attention blocks can effectively enhance DPCNN' s performance in classification tasks.

The second experiment in Section 4.3 evaluated DPDANet' s accuracy under different growth rates, finding that the largest growth rate is not necessarily optimal. Growth rate selection requires balancing performance improvement and computational efficiency to avoid negative impacts from either excessively small or large growth rates.

The third experiment in Section 4.3 on complexity found that introducing dense blocks into DPCNN can further reduce model parameter count and computational complexity. This is because dense connections achieve partial parameter sharing between layers, and the internal channels of stacked convolutional lay-

ers are narrow, effectively reducing model complexity. This experiment not only validates DPDANet' s advantages in deep semantic feature extraction and information flow but also indicates that rational network depth design is crucial for performance improvement, as excessive depth may lead to performance bottlenecks or even negative effects.

This paper proposes an improved DPCNN model—DPDANet—that integrates dense connections and attention mechanisms. Building upon DPCNN, dense connections enable each layer in the network to access feature information from all previous layers, thereby enhancing feature reuse capability and gradient flow and improving training stability. Simultaneously, introducing attention mechanisms into dense blocks allows the model to focus on semantic features of words that require attention, more effectively screening important information and reducing redundant information propagation. Extensive experiments demonstrate that the proposed model exhibits excellent performance in text classification tasks.

Future work will consider introducing heuristic algorithms (such as genetic algorithms or Bayesian optimization) to automatically search for optimal model parameter combinations, thereby reducing time spent on manual parameter tuning and further improving model robustness and performance. Through automated parameter optimization, we aim to more efficiently find optimal configurations for different datasets. Additionally, this paper plans to combine label embedding with contrastive learning methods in the embedding layer to better capture relationships between labels and input text, thereby further improving classification model accuracy and generalization capability.

References

[1] Dalal M K, Zaveri M A. Automatic Text Classification: A Technical Review[J]. International Journal of Computer Applications, 2011, 28: 37-40.
[2] Mikolov T, Chen K, Corrado G S, et al. Efficient Estimation of Word Representations in Vector Space[J/OL]. (2013-01-16). [2025-04-25]. https://arxiv.org/abs/1301.3781.
[3] Qi T, Wu F, Wu C, et al. HieRec: Hierarchical user interest modeling for personalized news recommendation[J]. arXiv preprint arXiv:2106.04408, 2021. [2021-06-10]. https://arxiv.org/abs/2106.04408.
[4] Liu X, Zheng L, Jia X, et al. Public opinion analysis on novel coronavirus pneumonia and interaction with event evolution in real world[J]. IEEE Transactions on Computational Social Systems, 2021, 8(4): 1042-1051.
[5] 李慧, 庞经纬. 基于文图音融合的多模态情感识别研究 *[J]. 数据分析与知识发现, 2024, 8(11): 11-21. (Li Hui, Pang Jingwei. Research on Multimodal Emotion Recognition Based on Text, Image, and Audio Fusion*[J]. Data Analysis and Knowledge Discovery, 2024, 8(11): 11-21.)
[6] Hossain M R, Hoque M M, Siddique N, et al. Bengali text document categorization based on very deep convolution neural network[J]. Expert Systems with Applications, 2021, 184: 115394.

[7] Duque A B, Santos L L J, Macêdo D, et al. Squeezed very deep convolutional neural networks for text classification[C]//In: Proceedings of the International Conference on Artificial Neural Networks. Cham: Springer International Publishing, 2019: 193-207.
[8] Johnson R, Zhang T. Deep pyramid convolutional neural networks for text categorization[C]//In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2017:
[9] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[J]. arXiv preprint arXiv:1301.3781, 2013.
[10] Pennington J, Socher R, Manning C. GloVe: Global Vectors for Word Representation[C]. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar. Stroudsburg: Association for Computational Linguistics, 2014: 1532-1543.
[11] Peters M E, Neumann M, Iyyer M, et al. Deep Contextualized Word Representations[C]. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), New Orleans, Louisiana, USA. Stroudsburg: Association for Computational Linguistics, 2018: 2227-2237.
[12] Devlin J, Chang M W, Lee K, et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding[C]. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, USA. Stroudsburg: Association for Computational Linguistics, 2019: 4171-4186.
[13] Achiam J, Adler S, Agarwal S, et al. GPT-4 Technical Report[J]. arXiv preprint arXiv:2303.08774, 2023.
[14] Yang Z, Dai Z, Yang Y, et al. XLNet: Generalized Autoregressive Pretraining for Language Understanding[J]. Advances in Neural Information Processing Systems, 2019, 32.
[15] 孙丹丹, 郑瑞坤. BERT-DPCNN 模型在网络舆情情感分析中的应用 [J]. 网络安全技术与应用, 2022(8): 24-27. (Sun Dandan, Zheng Ruikun. Application of BERT-DPCNN Model in Emotional Analysis of Network Public Opinion[J]. Network Security Technology & Application, 2022(8): 24-27.)
[16] 张静, 高子信, 丁伟杰. 基于 BERT-DPCNN 的警情文本分类研究 *[J]. 数据分析与知识发现, 2025, 9(2): 48-58. (Zhang Jing, Gao Zixin, Ding Weijie. Police Report Classification Based on BERT-DPCNN[J]. Data Analysis and Knowledge Discovery, 2025, 9(2): 48-58.)
[17] Kim Y. Convolutional Neural Networks for Sentence Classification[C]. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar. Association for Computational Linguistics, 2014: 1746-1751.
[18] Elman J L. Finding structure in time[J]. Cognitive Science, 1990, 14(2): 179-211.
[19] Graves A, Graves A. Long short-term memory[J]. Supervised Sequence Labelling with Recurrent Neural Networks, 2012: 37-45.
[20] Graves A, Schmidhuber J. Framewise phoneme classification with bidirec-

tional LSTM and other neural network architectures[J]. Neural Networks, 2005, 18(5-6): 602-610.
[21] Chung J, Gulcehre C, Cho K H, et al. Empirical evaluation of gated recurrent neural networks on sequence modeling[J]. arXiv preprint arXiv:1412.3555, 2014.
[22] Vaswani A, Shazeer N, Parmar N, et al. Attention is All You Need[J]. Advances in Neural Information Processing Systems, 2017, 30.
[23] Zhao Y, Tao X. ZYJ123@DravidianLangTech-EACL2021: Offensive language identification based on XLM-RoBERTa with DPCNN[C]. In: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, Kyiv, Ukraine. Kyiv: Association for Computational Linguistics, 2021: 216–221.
[24] Huang H, Yang K, Zhang L, et al. Intelligent Recommendation of Legal Articles Based on DPCNN with Capsule Model[C]. In: Proceedings of the 2021 International Conference on Computer Information Science and Artificial Intelligence (CISAI), Beijing, China. New York, USA: IEEE, 2021: 896-901.
[25] Xian G, Guo Q, Zhao Z, et al. Short Text Classification Model Based on DeBERTa-DPCNN[C]. In: Proceedings of the 2023 4th International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE), Zhangjiajie, China. New York, USA: IEEE, 2023: 56-59.
[26] Yu S, Liu D, Zhang Y, et al. DPTCN: A novel deep CNN model for short text classification[J]. Journal of Intelligent & Fuzzy Systems, 2021, 41(6): 7093-7100.
[27] Beck M, Pöppel K, Spanring M, et al. xlstm: Extended long short-term memory[J]. arXiv preprint arXiv:2405.04517, 2024.
[28] Huang G, Liu Z, Van Der Maaten L, et al. Densely connected convolutional networks[C]//In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017: 4700-4708.
[29] Zhang X, Zhao J, LeCun Y. Character-level convolutional networks for text classification[J]. Advances in Neural Information Processing Systems, 2015, 28.
[30] Socher R, Perelygin A, Wu J, et al. Recursive deep models for semantic compositionality over a sentiment treebank[C]. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, 2013: 1631-1642.
[31] Misra R. News Category Dataset [EB/OL]. [2022]. https://arxiv.org/abs/2209.11429.
[32] 杨森淇, 段旭良, 肖展, 等. 基于 ERNIE+DPCNN+BiGRU 的农业新闻文本分类 [J]. 计算机应用, 2023, 43(05): 1461-1466. (Yang Senqi, Duan Xuliang, Xiao Zhan, et al. Agricultural News Text Classification Based on ERNIE+DPCNN+BiGRU[J]. Journal of Computer Applications, 2023, 43(05): 1461-1466.)

Corresponding author: Dongmei Wang, ORCID: 0000-0002-4000-1601, E-mail: 20220467@m.scnu.edu.cn.

Funding: This work is supported by the Natural Science Foundation of Guangdong Province, China (Grant No. 2024A1515012126), and the College Students' Innovative Entrepreneurial Training Plan Program (Grant No. 202381005).

Author Contributions:
Luo Huayu: Drafted initial manuscript (Sections 3, 4, 5), visualization, validation, methodology, data preprocessing, conceptualization.
Wang Dongmei: Reviewed and edited manuscript, validation, supervision, project management.
Zhang Yinghui: Drafted initial manuscript (Sections 1, 2), visualization, literature review.
Lin Wei: Literature review, manuscript revision, data collection.
Chen Chen: Reviewed and edited manuscript.

Conflict of Interest Statement: All authors declare no conflict of interest.

Supporting Data:
[1] Luo Huayu. Douyin Comment Dataset.

Note: Figure translations are in progress. See original paper for figures.