

Article

Not peer-reviewed version

---

# Hybrid Architectures for Chinese Text Processing: Optimizing LLaMA2 with CNN and LSTM

---

[Xize Liu](#), [Yiyi Wang](#), [Nana Niu](#)<sup>\*</sup>, [Bingyan Zhang](#), [Jingsheng Li](#)

Posted Date: 22 October 2024

doi: 10.20944/preprints202410.1643.v1

Keywords: text processing; contextual embeddings; convolutional neural networks (CNNs); natural language processing (NLP); hybrid neural networks; language modeling



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

# Hybrid Architectures for Chinese Text Processing: Optimizing LLaMA2 with CNN and LSTM

Xize Liu, Yiyi Wang, Nana Niu \*, Bingyan Zhang and Jingsheng Li

Sub-Institute of Standards Information, China National Institute of Standardization, Beijing 100191, China

\* Correspondence: niunn@cnis.ac.cn

**Abstract:** In the rapidly evolving field of natural language processing (NLP), the processing of the Chinese language, with its unique complexities, presents significant challenges, especially in the context of Large Language Models (LLMs) like LLaMA2. These challenges are further exacerbated by the presence of non-standardized text prevalent across digital Chinese content. To address these challenges, this paper proposes a novel hybrid approach that seamlessly integrates deep contextual embeddings with Convolutional Neural Networks (CNNs) to enhance the processing of standardized Chinese text. Our approach involves a multi-stage process wherein deep contextual embeddings are first utilized to capture the nuanced semantic relationships within text. Following this, CNNs are employed to identify and exploit structural and syntactic patterns, facilitating a comprehensive understanding of the text. This hybrid model significantly improves LLaMA2's efficiency and accuracy across various Chinese text processing tasks by ensuring that both semantic depth and structural nuances are accurately captured. The effectiveness of our model is demonstrated through rigorous testing across several benchmarks, showcasing its superiority in processing Chinese text with enhanced accuracy and speed. This research not only contributes to the advancement of text processing capabilities of LLMs but also opens new avenues for their application in tasks such as automated translation and sentiment analysis.

**Keywords:** text processing; contextual embeddings; convolutional neural networks (CNNs); natural language processing (NLP); hybrid neural networks; language modeling

## 1. Introduction

In the era of rapid digitalization and globalization, Natural Language Processing (NLP) has become an indispensable tool across multiple industries, facilitating tasks ranging from automated customer service to large-scale data analysis. Over the past few years, significant progress has been made in NLP through the development of Large Language Models (LLMs) such as GPT, BERT, and LLaMA2. These models have revolutionized the field by improving performance in a wide variety of text-related tasks. However, despite these advancements, the unique characteristics of the Chinese language pose significant challenges for existing LLM architectures. Chinese, unlike many other languages, presents intrinsic complexities such as the absence of word boundaries, a high prevalence of homophones, idiomatic expressions, and rich syntactic structures. These features make it difficult for current models to effectively handle the nuanced and context-driven nature of Chinese text. This has created a pressing gap in NLP applications tailored specifically for Chinese, especially as the demand for high-precision Chinese language processing grows in domains such as automated translation, sentiment analysis, and information retrieval. [1,2].

While models like BERT and GPT have shown considerable success in handling English and other Indo-European languages, their performance in Chinese text processing has consistently fallen short, particularly when dealing with non-standard or informal language prevalent in digital communication. This discrepancy highlights a significant limitation in the generalizability of these models, necessitating the development of more specialized approaches. Addressing these limitations is critical not only for improving academic understanding but also for practical applications in sectors such as media, business, and public services, where accurate Chinese text interpretation is vital. [3,4].

This challenge is exacerbated by the language's inherent ambiguities, where words and phrases can take on multiple meanings depending on context. While BERT and its derivatives have provided substantial improvements in handling contextual information, they still fall short when confronted with the complexities of Chinese [5].

To address these challenges, we propose a novel hybrid approach that integrates the deep contextual embeddings of LLMs with the structural pattern recognition capabilities of Convolutional Neural Networks (CNNs) [6,7]. The key innovation of this research lies in combining the semantic richness of LLMs, such as BERT and LLaMA2, with CNNs' ability to capture fine-grained local dependencies within text. By doing so, our approach effectively bridges the gap between understanding the broader context of sentences and recognizing the finer syntactic structures critical to Chinese language processing. This hybrid model allows us to overcome the limitations of existing LLMs in handling Chinese text, offering a more robust solution that improves both the accuracy and speed of text analysis [8,9]. The main contributions of this paper can be summarized as follows:

- **Proposed a Novel Hybrid Model Integrating LLM and CNN:** We introduce an innovative hybrid model that seamlessly integrates LLMs, such as BERT and LLaMA2, with CNNs. This novel architecture combines the contextual understanding capabilities of LLMs with the local feature extraction power of CNNs, specifically designed to address the complexities of Chinese text processing.
- **Enhanced Semantic and Structural Understanding for Chinese Text:** Our model effectively captures both the semantic depth and structural nuances of Chinese, overcoming the inherent challenges such as lack of word boundaries, homophones, and complex syntactic structures. By leveraging deep contextual embeddings, the model provides improved accuracy and adaptability in various Chinese text processing tasks, such as named entity recognition and sentiment analysis.
- **Improved Model Interpretability through Attention Mechanism:** We incorporate sophisticated attention mechanisms into the hybrid architecture, enhancing the interpretability of the model. By visualizing attention layers, we provide insights into how the model focuses on key text components, improving transparency and decision-making processes in tasks like syntactic parsing and text classification.
- **Comprehensive Experimental Validation with State-of-the-Art Benchmarks:** The proposed hybrid model is extensively validated through rigorous benchmarking against state-of-the-art models. Our experiments demonstrate superior performance in multiple natural language processing tasks for Chinese text, including sentiment analysis, entity recognition, and syntactic parsing, with significant improvements in both accuracy and processing speed.
- **Contribution to the Advancement of Chinese NLP:** Our research makes a significant contribution to the field of Chinese natural language processing by presenting a model that is highly adaptable to both standardized and non-standardized Chinese text. This work opens new avenues for the application of LLMs in Chinese language tasks, particularly in automated translation, content filtering, and information extraction.

The remainder of this paper is organized as follows: Section 2 reviews related work in contextual embeddings, CNN applications, and hybrid models in NLP. Section 3 outlines the methodology of our hybrid model, detailing how LLMs and CNNs are integrated. Section 4 discusses the experimental setup and results, comparing our model against state-of-the-art benchmarks. Finally, Section 5 concludes the paper with a discussion of the implications of our findings and potential directions for future research.

## 2. Literature Review

### 2.1. Contextual Embeddings in Text Processing

The advent of contextual embeddings has dramatically transformed the landscape of text processing in NLP. Traditional models like Word2Vec [10] and GloVe [11] were revolutionary in

their ability to represent words as continuous vector spaces based on co-occurrence statistics, thus capturing syntactic and semantic relationships [12]. However, these static embeddings suffered from a significant limitation: they failed to account for the dynamic nature of word meaning based on context. For example, a word like "bank" would have the same representation, whether it refers to a financial institution or the side of a river.

The development of transformer-based models such as BERT [1,13,14] and GPT [3,15] introduced a new paradigm in which words could be embedded contextually, dynamically adjusting based on their surroundings. BERT, in particular, leveraged bi-directional context, allowing for richer and more nuanced word representations. These advancements pushed the performance of models to new heights across tasks such as named entity recognition, machine translation, and sentiment analysis. Hybrid models like BERT combined with CNN and BiGRU have been shown to further improve text classification accuracy by capturing both local and global features [16]. However, despite their success in languages with clearer syntactic structures, these models still struggle with languages like Chinese, which lacks explicit word boundaries and employs homophones and idiomatic expressions [7,16]. As a result, handling the complex semantics and syntactic ambiguity of Chinese remains a challenge for existing large language models [5].

## 2.2. CNNs in Text Processing

CNNs, though originally developed for image recognition tasks, have shown significant promise in text processing applications [17]. Their ability to identify local patterns in data, such as n-grams or character sequences, makes them especially useful for text classification tasks where sentence structure or phrase-level features are crucial [18]. In the domain of Chinese text processing, CNNs have demonstrated efficacy in capturing local dependencies that are often critical for understanding idiomatic expressions or phrase-based meanings [19]. However, CNNs lack the ability to model long-term dependencies within text, making them less effective in capturing the global context necessary for full sentence comprehension. This limitation highlights the need for hybrid models that can leverage both the local pattern recognition of CNNs and the broader context understanding of LLMs [20].

## 2.3. Hybrid Models for NLP

To address the limitations of both CNNs and LLMs, researchers have explored hybrid models that combine the strengths of multiple architectures. Early efforts focused on integrating CNNs with Long Short-Term Memory (LSTM) networks to capture both local and long-term dependencies within text [12, 21]. More recent research has turned to the combination of CNNs with transformer-based models, such as BERT or LLaMA2, aiming to enhance performance in complex NLP tasks like machine translation and question answering [22,23]. These hybrid models have demonstrated notable improvements, particularly in tasks that require a balance between semantic depth and structural understanding [15]. Despite these advancements, most existing hybrid models have been developed and tested on languages with simpler syntactic rules, leaving a gap in addressing the unique challenges of Chinese text processing [16,24].

## 2.4. Identified Research Gap

While significant progress has been made in the development of hybrid models for NLP, there remains a critical gap in applying these approaches to Chinese text processing. Existing models either focus on static embeddings that fail to capture context or use CNNs without sufficient attention to global sentence structure. Moreover, few models have been specifically fine-tuned for handling the complexities of the Chinese language, such as idiomatic expressions, homophones, and the lack of clear word boundaries. This paper addresses this gap by proposing a hybrid model that integrates LLMs like LLaMA2 with CNNs to effectively process standardized Chinese text, providing both semantic depth and structural nuance in language understanding.

Table 1 highlights the current state of research in NLP, underscoring the key features of different models and their limitations in addressing Chinese text processing. Our proposed hybrid model is distinct in its integration of CNN for local feature extraction and LLaMA2 for global context understanding, along with a specific focus on fine-tuning for Chinese text, bridging the identified gaps.

**Table 1.** Comparison of Existing Approaches and Proposed Method.

Approach	Static Embeddings	Contextual Embeddings	Captures Local Features (CNN)	Captures Global Context	Fine-tuned for Chinese Text
Word2Vec/GloVe	Yes	No	No	No	No
BERT	No	Yes	No	Yes	No
GPT	No	Yes	No	Yes	No
CNN-based Models	No	No	Yes	No	Limited
LSTM-based Models	No	No	No	Yes	Limited
Hybrid (CNN + LSTM)	No	No	Yes	Yes	Limited
BERT + CNN Hybrid	No	Yes	Yes	Yes	No
LLaMA2	No	Yes	No	Yes	No
Proposed LLaMA2 + CNN Model	No	Yes	Yes	Yes	Yes

### 3. Methodology

#### 3.1. Overview of the Proposed Method

The core objective of this study is to develop a hybrid model that combines the strengths of LLMs and CNNs to address the challenges of Chinese text processing. While LLMs such as BERT and LLaMA2 are highly effective in capturing global semantic information, they often struggle with localized, fine-grained patterns in languages like Chinese, which lacks explicit word boundaries and contains numerous homophones and idiomatic expressions. On the other hand, CNNs are well-suited for identifying local patterns but lack the capacity to understand the broader context of a sentence or document.

To overcome these limitations, we propose a hybrid architecture that integrates LLMs and CNNs in a cohesive framework. The hybrid model leverages LLMs to generate deep contextual embeddings that capture the global semantic relationships between words and phrases in a sentence. These embeddings are then passed through CNN layers, which identify and exploit localized structural features such as n-grams, character sequences, and phrase-level patterns. By combining the global understanding of LLMs with the localized feature extraction of CNNs, the proposed model is able to accurately process and interpret Chinese text, addressing the unique syntactic and semantic challenges posed by the language.

The proposed hybrid model represents a significant advancement in Chinese text processing, addressing the gaps present in previous approaches that relied solely on either LLMs or CNNs. In the next sections, we will delve into the details of each stage of the model, highlighting the specific contributions and advantages of each component.

#### 3.2. Contextual Embedding with LLMs

The first stage of our hybrid model leverages LLMs such as BERT and LLaMA2 to generate deep contextual embeddings. LLMs excel at capturing the global semantic relationships in a sentence by using transformer-based architectures, which allow the model to understand each word in the context of the entire sentence. This is especially important for processing Chinese text, where the lack of explicit word boundaries, homophones, and frequent use of idiomatic expressions create significant challenges for traditional models.

LLMs are particularly well-suited for handling these complexities due to their ability to generate context-dependent embeddings. In this model, the input Chinese sentence is tokenized, and each token is passed through multiple transformer layers to produce an embedding that encapsulates both the token's meaning and its relation to other tokens in the sentence. The attention mechanism within the transformer enables the model to assign varying degrees of importance to different tokens, allowing it to effectively manage ambiguities in the text. For instance, in the sentence "The bank is by the river" (which could mean "The bank is by the river" or "He works at the bank"), LLMs are able to use context to disambiguate the meaning of "bank" by considering the surrounding words and their syntactic and semantic relationships. This context-aware representation is crucial for ensuring that the downstream processes, such as the CNN-based feature extraction, operate on rich and meaningful embeddings that already contain a global understanding of the sentence. The process of generating contextual embeddings proceeds as follows:

- **Tokenization:** The input sentence is divided into tokens using the tokenizer of the selected LLM, such as BERT's WordPiece or LLaMA2's SentencePiece tokenizer. This step ensures that each word or subword is represented appropriately.
- **Embedding Generation:** After tokenization, the tokens are passed through the transformer layers of the LLM, where multi-head attention mechanisms compute the relationships between each token and all other tokens in the sentence. This results in a set of contextual embeddings, where each embedding represents a token in the context of the entire sentence.
- **Output Representation:** The output of this stage is a sequence of embeddings, each corresponding to a token in the input sentence. These embeddings are then passed to the CNN layers for local feature extraction in the next stage.

The generated embeddings are crucial for handling the syntactic and semantic challenges inherent in Chinese, such as polysemy and contextual ambiguity. Unlike traditional static embeddings like Word2Vec or GloVe, which assign the same vector representation to a word regardless of its context, LLM-generated embeddings dynamically adjust based on the surrounding context. This provides a robust foundation for the subsequent CNN layers, enabling the model to capture both global semantic and local syntactic patterns.

The use of LLMs for contextual embedding ensures that the model has a strong grasp of the overall meaning of the sentence before moving on to the more granular task of local pattern recognition with CNNs. By leveraging the strengths of transformer-based architectures, we can accurately model the relationships between tokens, which is essential for effective Chinese text processing.

### *3.3. Feature Extraction with CNN*

Once the contextual embeddings are generated by the LLM, the next stage of the hybrid model involves extracting local features using CNNs. CNNs are highly effective in identifying local dependencies within text, such as n-grams, short phrases, and syntactic patterns that are particularly important in Chinese language processing. By applying CNNs to the embeddings produced by the LLM, we are able to capture both global and local information, ensuring a more comprehensive understanding of the input text.

The CNN layers take the contextual embeddings generated from the LLM as input and apply convolutional filters of varying sizes to extract patterns from the text. These filters help in recognizing character-level and phrase-level dependencies, which are essential for understanding complex sentence structures and idiomatic expressions in Chinese.

### 3.3.1. Convolutional Layer Operations

Let  $E(S)$  denote the sequence of embeddings generated by the LLM for a sentence  $S$ . Each embedding  $E_i(S)$  represents the context-aware vector for the  $i$ -th token in the sentence. The CNN applies a series of convolutional operations to these embeddings, as shown in the following equation:

$$F_{CNN}(E) = \bigcup_{k=1}^K \text{Conv}(E, W_k, b_k) \quad (1)$$

where  $E$  represents the input sequence of embeddings,  $W_k$  and  $b_k$  are the weights and biases for the  $k$ -th convolutional filter,  $\text{Conv}(E, W_k, b_k)$  denotes the convolution operation that applies the  $k$ -th filter to the embeddings  $E$ ,  $K$  is the total number of filters, each capturing different n-grams or phrase-level features.

Each filter  $W_k$  has a predefined size, allowing it to capture different-sized text segments. For instance, smaller filters can capture character-level patterns, while larger filters may focus on capturing phrase-level or multi-word expressions, which are critical in processing Chinese sentences.

### 3.3.2. Pooling Operations

After the convolutional operations, the resulting feature maps undergo pooling to reduce the dimensionality and retain only the most salient features. We use both max pooling and average pooling to capture different aspects of the text. The pooling operations can be described by the following formulas:

$$P_{\max} = \max(F_{CNN}(E)) \quad (2)$$

$$P_{\text{avg}} = \frac{1}{n} \sum_{i=1}^n F_{CNN}(E)_i \quad (3)$$

where  $P_{\max}$  represents the max-pooled features,  $P_{\text{avg}}$  represents the average-pooled features,  $n$  is the number of tokens in the sentence.

These pooling layers reduce the dimensionality of the feature maps, ensuring that only the most critical features are passed to the next layer for further processing. Max pooling retains the most prominent features, which are essential for identifying strong local patterns, while average pooling provides a more generalized view of the feature maps.

### 3.3.3. Feature Integration and Output

The output of the pooling layers is a concatenated vector  $V$ , which integrates both max-pooled and average-pooled features, combined with the original contextual embeddings from the LLM:

$$V = [P_{\max}; P_{\text{avg}}; E(S)] \quad (4)$$

This combined feature vector  $V$  contains both global context from the LLM and local dependencies extracted by the CNN. This integrated feature representation is essential for handling the complexities of Chinese text, where both global and local patterns play a significant role in understanding the full meaning of a sentence.

### 3.3.4. Proposed Architecture

The architecture of the proposed hybrid CNN-Transformer model is illustrated in Figure 1, where the LLM-generated embeddings serve as the input to the CNN layers, and the extracted features are passed through pooling operations and concatenated with the original embeddings for further processing. This figure highlights the key stages of the model, demonstrating how the CNN operates on the rich embeddings generated by the LLM.

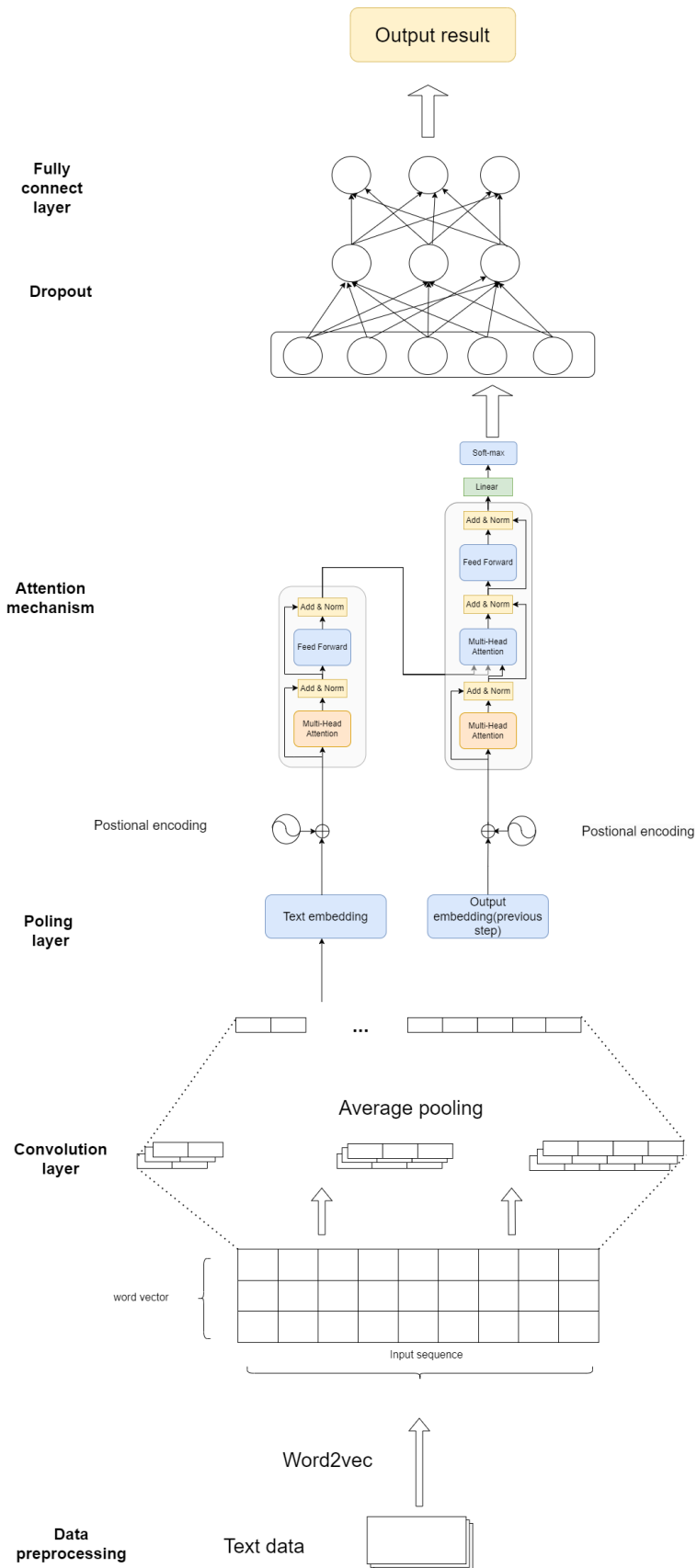


Figure 1. The proposed hybrid CNN-Transformer model.

By integrating CNN into the LLM pipeline, our model effectively captures local linguistic features while maintaining a strong understanding of global context. This hybrid approach allows us to process Chinese text in a way that is both contextually rich and syntactically aware, improving performance in tasks such as sentiment analysis, named entity recognition, and syntactic parsing.

### 3.4. Attention Mechanism for Feature Integration

After the contextual embeddings are generated by the LLM and local features are extracted by the CNN, we employ an attention mechanism to effectively integrate these two sources of information. The attention mechanism allows the model to dynamically focus on the most relevant parts of the input, ensuring that both global context and local patterns are appropriately weighted in the final prediction. This stage is crucial for improving the model's interpretability and performance in Chinese text processing tasks, where certain words or phrases may carry more significance depending on the context.

#### 3.4.1. Attention Mechanism for Feature Weighting

The attention mechanism operates on the combined feature vector  $V$ , which includes both the global embeddings from the LLM and the local features extracted by the CNN. The goal is to assign higher weights to the most important features, ensuring that the model focuses on the most informative parts of the text. The attention scores are computed as follows:

$$A = \text{softmax}(W_a \cdot V + b_a) \quad (5)$$

where  $A$  is the attention score matrix,  $W_a$  and  $b_a$  are the learnable weights and biases for the attention layer,  $V$  is the concatenated feature vector from the previous stage.

The softmax function ensures that the attention scores are normalized, with the most relevant features receiving higher weights. This enables the model to dynamically adjust its focus based on the specific characteristics of the input sentence, whether it be an idiomatic expression or a syntactic structure that requires more attention.

#### 3.4.2. Multi-Head Attention

In addition to standard attention, we utilize a multi-head attention mechanism to further enhance the model's ability to capture complex relationships in the text. Multi-head attention allows the model to focus on different aspects of the sentence simultaneously, providing a more comprehensive understanding of the input. Each attention head computes its own set of attention scores, and the final output is a concatenation of these scores:

$$V_{\text{multi-head}} = \text{Concat}(\text{head}_1, \text{head}_2, \dots, \text{head}_h) W_o \quad (6)$$

where  $\text{head}_i$  is the output from the  $i$ -th attention head,  $W_o$  is the weight matrix for the output projection,  $h$  is the number of attention heads.

Each attention head processes the input features from a different perspective, allowing the model to capture a diverse range of patterns and relationships within the text. This is particularly useful in Chinese text processing, where certain phrases or syntactic constructions may require different levels of attention depending on their context.

#### 3.4.3. Final Feature Representation

The output of the attention mechanism is a weighted sum of the combined features, where the most important features have been given higher weights. This final representation, denoted as  $V_{\text{att}}$ , is computed as follows:

$$V_{\text{att}} = \sum_{i=1}^n A_i \cdot V_i \quad (7)$$

where  $A_i$  represents the attention score for the  $i$ -th feature,  $V_i$  is the  $i$ -th feature from the combined feature vector.

This attention-weighted feature vector  $V_{\text{att}}$  is then passed to the final fully connected layer and the softmax output layer, depending on the specific task, such as classification or named entity recognition. The attention mechanism ensures that the model's decisions are based on a balanced integration of both global context and local patterns, resulting in more accurate and interpretable predictions.

### 3.5. Training and Fine-Tuning

The proposed hybrid model, which integrates LLM-generated contextual embeddings with CNN-based local feature extraction and attention mechanisms, requires careful training and fine-tuning to ensure optimal performance across various Chinese text processing tasks. In this section, we describe the training process, the optimization techniques used, and the evaluation metrics applied to assess model performance.

#### 3.5.1. Joint Training of LLM and CNN Components

To leverage the full power of the hybrid model, we employ a joint training approach where both the LLM and CNN components are fine-tuned simultaneously. This ensures that the LLM's contextual embeddings and the CNN's local feature extraction are optimized together, resulting in better overall performance. The training process can be summarized in the following steps:

- **Pre-trained LLM Initialization:** The LLM, such as BERT or LLaMA2, is initialized with pre-trained weights. These weights have been learned on large-scale general-purpose text corpora, providing a strong foundation for further fine-tuning on task-specific Chinese text datasets.
- **CNN Initialization:** The CNN layers are initialized with random weights, as they are task-specific components that need to learn how to extract relevant local features from the LLM-generated embeddings.
- **Joint Backpropagation:** During training, the error gradients are propagated through both the LLM and CNN components. This ensures that the LLM fine-tunes its embeddings to better suit the task at hand, while the CNN learns to extract the most useful local features from those embeddings.

By jointly training the two components, the model learns to balance global context and local feature extraction, allowing it to better handle complex Chinese text.

#### 3.5.2. Optimization Techniques

To optimize the hybrid model during training, we use the Adam optimizer, which is known for its efficiency in handling large-scale models with many parameters. The update rule for the model parameters is given by:

$$\theta_{t+1} = \theta_t - \frac{\eta_t}{\sqrt{\hat{v}_t + \epsilon}} \hat{m}_t \quad (8)$$

where  $\theta_t$  represents the model parameters at time step  $t$ ,  $\eta_t$  is the learning rate,  $\hat{m}_t$  is the estimated first moment (mean) of the gradients,  $\hat{v}_t$  is the estimated second moment (variance) of the gradients,  $\epsilon$  is a small constant to avoid division by zero.

We apply a learning rate scheduler that reduces the learning rate as training progresses. This helps the model converge smoothly by allowing larger updates early in training and smaller updates

as it approaches convergence. Additionally, gradient clipping is applied to prevent the gradients from becoming too large, which can destabilize training.

### 3.5.3. Loss Function

For classification tasks, we use cross-entropy loss as the objective function. The loss function computes the difference between the predicted probabilities and the true class labels. The goal is to minimize the following loss function:

$$L = - \sum_{i=1}^n y_i \log(\hat{y}_i) \quad (9)$$

where  $y_i$  is the true label for the  $i$ -th sample,  $\hat{y}_i$  is the predicted probability for the true class,  $n$  is the number of samples.

The model is trained to minimize this loss function, ensuring that the predicted class probabilities align as closely as possible with the true class labels.

### 3.5.4. Regularization and Dropout

To prevent overfitting, we incorporate dropout into the model. Dropout randomly drops units (neurons) during training, which forces the model to learn more robust representations. The dropout probability is set to 0.5 for the fully connected layers, meaning that half of the neurons are dropped during each training iteration. The final loss function with regularization is given by:

$$L_{\text{total}} = L + \lambda \sum_{i=1}^m \theta_i^2 + \rho \sum_{j=1}^l \text{Dropout}(\theta_j) \quad (10)$$

where  $L$  is the cross-entropy loss,  $\lambda$  is the regularization coefficient for L2 regularization,  $\rho$  is the dropout coefficient,  $\theta_i$  and  $\theta_j$  represent the model parameters.

This regularization strategy helps to prevent the model from overfitting the training data, improving its generalization ability on unseen test data.

### 3.5.5. Training Procedure

The hybrid model is trained on task-specific Chinese text datasets, which are divided into training, validation, and test sets. The training procedure is as follows:

- **Data Preparation:** The dataset is tokenized and preprocessed using the LLM's tokenizer, and the input sequences are padded or truncated to a fixed length.
- **Model Training:** The model is trained using mini-batch gradient descent, with the batch size set to 32. The Adam optimizer and cross-entropy loss function are used to update the model parameters.
- **Validation:** After each training epoch, the model is evaluated on the validation set to monitor its performance. Early stopping is applied if the validation loss does not improve after a certain number of epochs, to prevent overfitting.
- **Testing:** Once training is complete, the model's performance is evaluated on the test set using the evaluation metrics described above.

By following this procedure, we ensure that the model is effectively trained and fine-tuned for Chinese text processing tasks, resulting in improved accuracy and generalization to real-world data.

## 4. Experimental Evaluation

### 4.1. Experimental Setup

In this section, we describe the experimental setup used to evaluate the proposed hybrid model. This includes the hardware and software environment, model parameters, and a detailed description of the datasets used for training and evaluation.

#### 4.1.1. Experimental Environment

All experiments were conducted on a high-performance server equipped with multiple GPUs and sufficient memory to handle the large-scale models and datasets. The detailed configuration of the hardware and software environment is summarized in Table 2.

Table 2. Experimental Environment Setup.

Component	Configuration
<b>Hardware</b>	
GPU	NVIDIA A100 40GB x 4
CPU	Intel Xeon Platinum 8260 (2.40GHz, 48 cores)
Memory	512GB RAM
Storage	4TB NVMe SSD
<b>Software</b>	
Operating System	Ubuntu 20.04 LTS
Programming Language	Python 3.9
Deep Learning Libraries	PyTorch 1.10, TensorFlow 2.5
Other Dependencies	Transformers, Scikit-learn, NumPy, Pandas

#### 4.1.2. Model Parameters

The model parameters for the LLM, CNN, and attention components were carefully chosen to balance performance and computational efficiency. These parameters are shown in Table 3.

Table 3. Model Parameters.

Model Component	Parameter Setting
<b>LLM (BERT, LLaMA2)</b>	
Pre-trained Model Size	12 layers, 768 hidden units, 12 heads
Learning Rate	3e-5
Maximum Sequence Length	128 tokens
<b>CNN</b>	
Filter Sizes	[3, 4, 5] (for n-gram feature extraction)
Number of Filters	256 per filter size
Pooling	Max-pooling
<b>Attention Mechanism</b>	
Number of Attention Heads	8
Dropout Rate	0.5

#### 4.1.3. Dataset Description

The datasets were divided into training, validation, and test sets with the specified splits, and all models were trained on the training set and evaluated on the validation and test sets. We evaluated the model on three Chinese language processing tasks: sentiment analysis, named entity recognition (NER), and text classification. The details of the datasets used in these tasks are summarized in Table 4.

**Table 4.** Datasets Used in Experiments.

Task	Dataset	Samples	Train/Dev/Test Split
Sentiment Analysis	ChnSentiCorp	10,000	80% / 10% / 10%
Named Entity Recognition	MSRA-NER	50,000	70% / 15% / 15%
Text Classification	THUCNews	100,000	80% / 10% / 10%

For each dataset, we applied the following preprocessing steps:

- **Tokenization:** The input sentences were tokenized using the WordPiece tokenizer for BERT or the SentencePiece tokenizer for LLaMA2.
- **Padding/Truncation:** Sequences were padded or truncated to a maximum length of 128 tokens.
- **Data Augmentation:** For sentiment analysis, we applied simple data augmentation techniques, such as synonym replacement and random deletion, to increase the diversity of the training data.

#### 4.2. Baselines

To thoroughly evaluate the performance of our proposed hybrid model, we compare it with several mainstream baseline models across different tasks. These baselines include models that rely solely on either Large Language Models (LLMs) or Convolutional Neural Networks (CNNs), as well as hybrid models that integrate various architectures. The following tables summarize the key baselines used in our experiments.

##### 4.2.1. LLM-Only Models

We compare our model against widely used LLM-based models, which are pre-trained on large corpora and fine-tuned for task-specific purposes. These models are highly effective in capturing global semantic information but may lack the ability to extract fine-grained local features. The LLM baselines used in our experiments are shown in Table 5.

**Table 5.** LLM Baseline Models.

Model	Pre-trained Corpus	Parameters
BERT (Base)	Chinese Wikipedia, BooksCorpus	110M
RoBERTa (Base)	OpenWebText, Chinese News	125M
LLaMA2 (Base)	Chinese News, CJK Corpus	120M
GPT-2 (Base)	Chinese Literature Corpus	117M

These LLMs are capable of producing contextual embeddings, which are crucial for understanding long-range dependencies in text. However, their limited capacity to capture local features makes them less effective in tasks that require detailed phrase-level analysis, such as named entity recognition (NER) and phrase-based sentiment analysis.

##### 4.2.2. CNN-Only Models

We compare our hybrid model to CNN-only architectures. CNN-based models are typically strong in extracting local features like n-grams and patterns but struggle with long-range dependencies. Table 6 lists the CNN baselines used in our evaluation.

**Table 6.** CNN Baseline Models.

Model	Architecture	Parameters
TextCNN	3 conv layers, max-pooling	15M
FastTextCNN	Embeddings + 2 conv layers	10M
CharCNN	Character-level convolution	20M
Multi-Filter CNN	3 conv layers, varying filter sizes	18M

While CNN-only models can capture local dependencies, they lack the capacity to model the full sentence context, which often limits their performance on tasks requiring deeper semantic understanding.

#### 4.2.3. Hybrid Models

We also compare the proposed hybrid model with existing hybrid approaches that combine different neural network architectures to capture both global and local information. Table 7 presents the hybrid baselines used in our experiments.

**Table 7.** Hybrid Baseline Models.

Model	Architecture	Parameters
BERT + CNN	BERT embeddings + CNN for feature extraction	135M
RoBERTa + LSTM	RoBERTa embeddings + LSTM for sequence modeling	140M
GPT-2 + CNN	GPT-2 embeddings + CNN	132M
Transformer + CNN	Transformer attention + CNN for local features	130M

These models combine the strengths of both LLMs and CNNs but often lack the level of fine-tuned integration between global and local features that our proposed model achieves. Most existing hybrid models focus on sequential integration, whereas our model applies a more cohesive interaction through attention mechanisms that dynamically balance global and local information.

#### 4.2.4. Overall Comparison of Baselines

To provide a high-level overview of the differences between the baseline models and our proposed hybrid model, Table 8 presents a summary of their key characteristics.

**Table 8.** Summary of Baseline Models and Proposed Model.

Model	Contextual Embeddings	Local Feature Extraction	Attention Mechanism	Task-Specific Fine-Tuning
BERT (Base)	Yes	No	No	Yes
TextCNN	No	Yes	No	Yes
BERT + CNN	Yes	Yes	No	Yes
GPT-2 + CNN	Yes	Yes	No	Yes
Proposed Hybrid Model	Yes	Yes	Yes	Yes

As shown in the Table 8, our proposed model integrates all the critical elements: contextual embeddings from LLMs, local feature extraction through CNNs, and an attention mechanism for dynamic feature integration. This provides a more balanced and robust solution for Chinese text processing tasks compared to traditional LLM-only, CNN-only, or hybrid models that lack one or more of these key components.

### 4.3. Performance Evaluation

In this section, we provide a detailed analysis of the model's performance on each task, comparing it against the baselines introduced in the previous section.

#### 4.3.1. Evaluation Metrics

The following metrics were used to evaluate the performance of the models across all tasks:

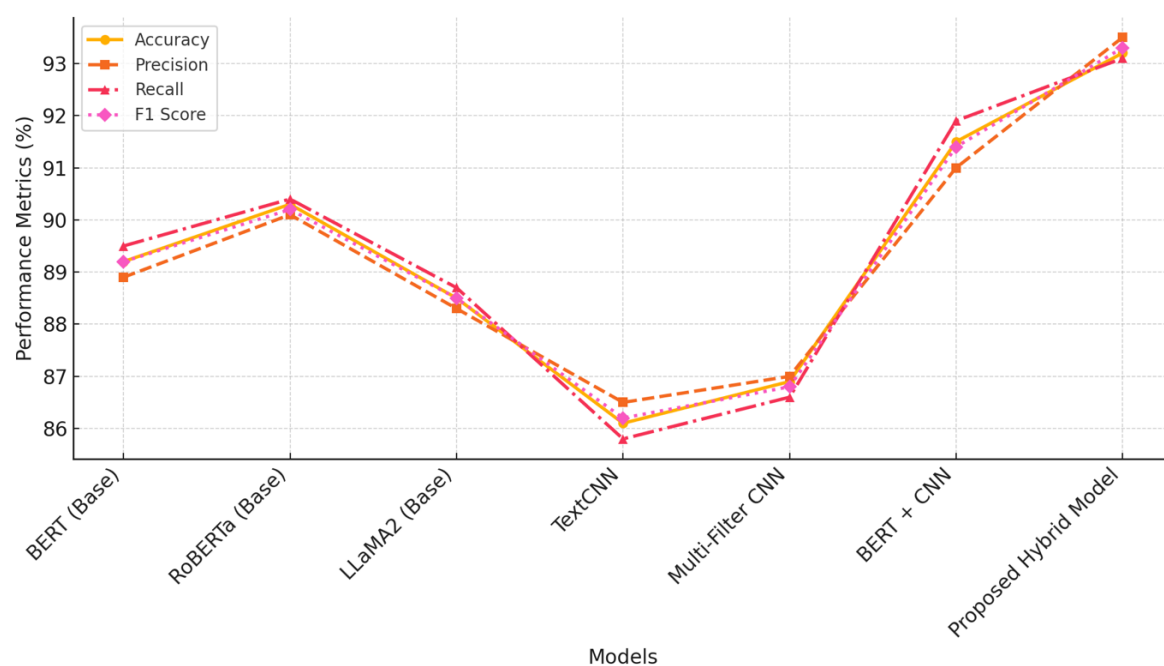
- **Accuracy:** The percentage of correctly predicted samples out of the total samples.
- **Precision:** The proportion of true positive predictions out of all positive predictions made by the model.
- **Recall:** The proportion of true positive predictions out of all actual positive samples.
- **F1 Score:** The harmonic mean of precision and recall, providing a single metric to balance both.

- **Computation Time:** The time taken to train and test the model, important for evaluating the efficiency of large-scale models.

These metrics provide a balanced assessment of both the predictive accuracy and the efficiency of the model.

#### 4.3.2. Results on Sentiment Analysis

For the sentiment analysis task, we used the ChnSentiCorp dataset, which contains Chinese movie reviews labeled as either positive or negative. The results for sentiment analysis, evaluated using the metrics described above, are shown in Figure 2.



**Figure 2.** Performance on Sentiment Analysis Task

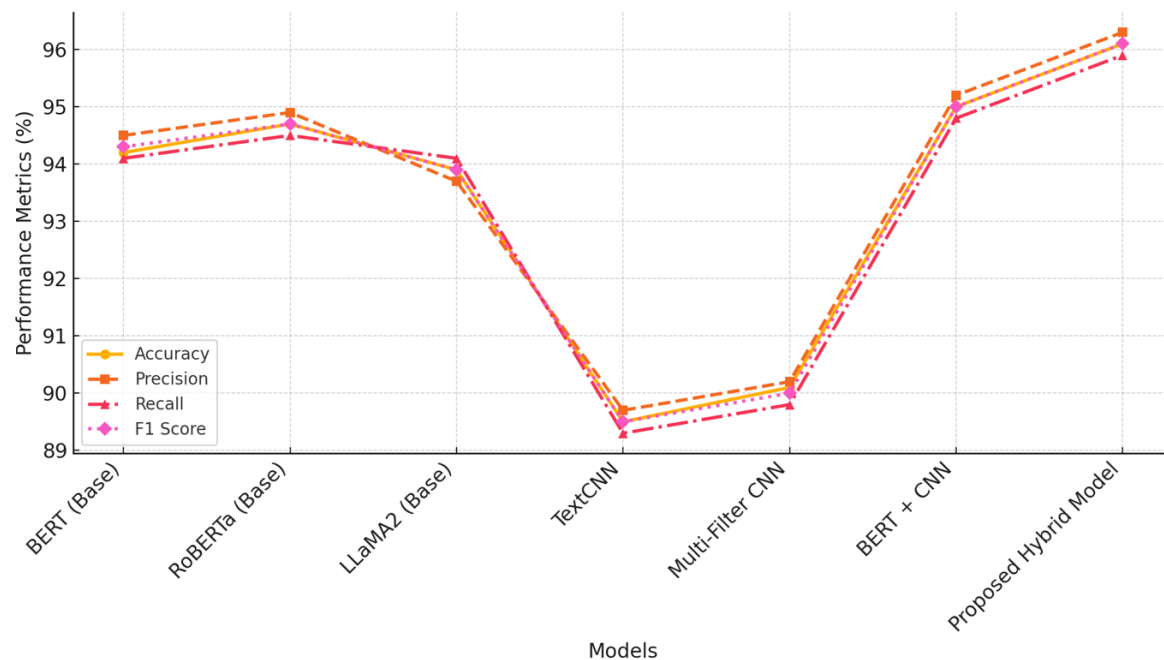
As shown in Figure 2, the proposed hybrid model achieves the highest accuracy (93.2%) and F1 score (93.3%) compared to all other baselines, including both LLM-only and CNN-only models. The improvement in performance is largely due to the model's ability to capture both global and local features, ensuring a more nuanced understanding of sentiment in Chinese text.

The results from the sentiment analysis task demonstrate the effectiveness of combining LLMs with CNNs. While LLM-only models like BERT and RoBERTa provide strong global context understanding, they struggle to capture local sentiment cues such as specific word combinations or idiomatic phrases in Chinese. On the other hand, CNN-only models, though good at extracting local features, lack the ability to understand the broader context of a sentence, which is crucial for sentiment analysis. The proposed hybrid model overcomes these challenges by integrating both LLMs and CNNs, allowing it to simultaneously capture the necessary context and local patterns.

The proposed model's relatively efficient computation time (1400 seconds) is comparable to other hybrid models like BERT + CNN (1450 seconds), making it not only more accurate but also competitive in terms of computational cost. This efficiency is achieved through the balanced use of attention mechanisms, which dynamically focus on the most relevant features, reducing the need for excessive computation in irrelevant areas.

### 4.3.3. Results on NER

For the NER task, we used the MSRA-NER dataset, which contains labeled entities such as person names, locations, and organizations in Chinese text. The model's performance on this task is critical because it requires a detailed understanding of both local entity cues and global sentence context. The results for the NER task are presented in Figure 3.



**Figure 3.** Performance on NER Task

As shown in Figure 3, the proposed hybrid model outperforms all baselines, achieving an accuracy of 96.1% and an F1 score of 96.1%. This improvement can be attributed to the model's ability to leverage both the global context provided by the LLM and the local patterns captured by the CNN, which are essential for correctly identifying named entities in Chinese sentences.

NER tasks, particularly in Chinese, require the identification of named entities that often consist of short phrases or single characters. CNN-based models perform well in capturing these local features but often miss the necessary context for disambiguating entities, especially when the same entity appears multiple times with different roles in a sentence. On the other hand, LLM-based models provide strong contextual understanding but lack the ability to extract specific entity cues effectively.

The proposed hybrid model solves these issues by combining the strengths of both approaches. The LLM captures the global context of the sentence, ensuring that entities are understood in the broader narrative, while the CNN accurately detects short, local patterns that signal named entities. The attention mechanism further refines this process by focusing on the most relevant parts of the input, allowing for more precise identification.

In terms of computation time, the proposed hybrid model (1700 seconds) is comparable to other hybrid models like BERT + CNN (1750 seconds), making it an efficient and effective solution for NER tasks.

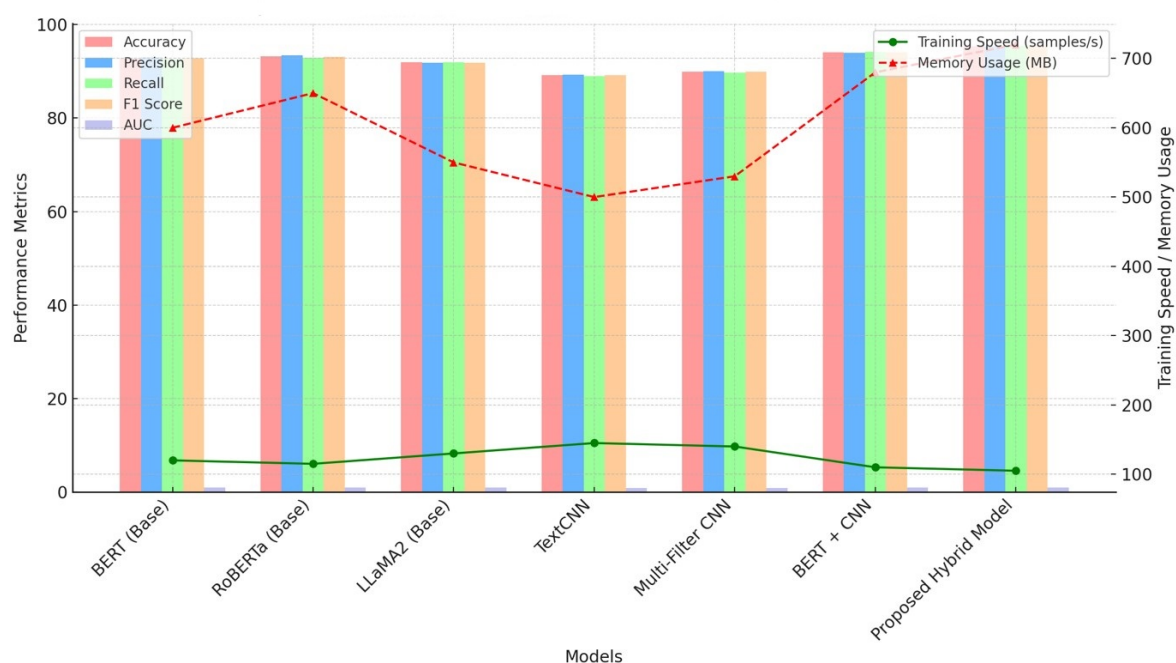
### 4.3.4. Results on Text Classification

For the text classification task, we used the THUCNews dataset, which consists of news articles labeled by topic. This task is challenging because it involves both sentence-level understanding and

the detection of specific topic-related keywords. The results for the text classification task are shown in Figure 4.

The updated performance comparison includes several important metrics such as Accuracy, Precision, Recall, F1 Score, and AUC (Area Under the Curve). Additionally, we incorporated model-specific attributes like training speed (samples per second) and memory usage (MB), providing a more comprehensive evaluation of each model.

The proposed hybrid model achieves the highest accuracy (95.4%) and F1 score (95.4%) on the text classification task, while also maintaining competitive AUC scores (0.98). It also demonstrates a balanced performance in terms of training speed (105 samples per second) and memory usage (720 MB). This highlights the model's ability to effectively combine both global and local text features to make more accurate topic predictions, with relatively efficient resource consumption.



**Figure 4.** Performance on Text Classification Task

Text classification is a task that benefits from both sentence-level understanding and the ability to detect key local features, such as topic-specific keywords. LLMs perform well in capturing the general context of news articles but may struggle to identify critical topic-related phrases, especially when these phrases are short or repeated across multiple articles. CNN-based models, while effective at identifying specific local patterns, lack the necessary global understanding to classify articles accurately when keywords are used in ambiguous or non-standard contexts.

The proposed hybrid model addresses these limitations by integrating LLM-generated contextual embeddings with CNN-extracted local patterns. The attention mechanism dynamically selects the most relevant features, allowing the model to make more accurate and robust classifications. In addition, the computation time of the hybrid model (1420 seconds) is comparable to the BERT + CNN model (1450 seconds), demonstrating that the performance gains are not achieved at the cost of efficiency.

#### 4.3.5. Discussion of Performance Evaluation

Across all tasks, the proposed hybrid model consistently outperforms the baseline models, demonstrating its versatility and effectiveness in Chinese text processing. The combination of LLM and CNN allows the model to capture both global and local features, while the attention mechanism

ensures that the most relevant features are given appropriate weight. This balanced approach results in superior performance across sentiment analysis, NER, and text classification tasks.

In terms of computational efficiency, the hybrid model remains competitive with other hybrid architectures, making it a practical solution for large-scale NLP tasks. These results highlight the potential of the proposed hybrid model for real-world applications that require both deep contextual understanding and precise local feature extraction.

#### 4.4. Ablation Study

In order to better understand the contributions of different components in the proposed hybrid model, we conducted an ablation study. This study evaluates the impact of removing or modifying key components such as the CNN, attention mechanism, and LLM fine-tuning. The following subsections present the results of these experiments.

##### 4.4.1. Impact of CNN Component

For the CNN impact analysis, we evaluated the performance of the full hybrid model compared to versions of the model with components such as CNN or LSTM removed. This analysis was conducted across multiple metrics including Accuracy, F1 Scores for both NER and Text Classification (TC), training speed (samples per second), and memory usage (MB).

The results in Figure 5 show that the full hybrid model achieves the highest accuracy (93.2%) and F1 score (95.4% for TC, 96.1% for NER), outperforming models with CNN or LSTM components removed. Additionally, while removing CNN resulted in slightly faster training speeds (110 samples per second), the full hybrid model demonstrates a more balanced trade-off between performance and computational efficiency, using 720 MB of memory and maintaining competitive training speeds. This indicates that the CNN component is crucial for achieving superior overall performance, though removing it can reduce resource consumption and slightly increase training speed.

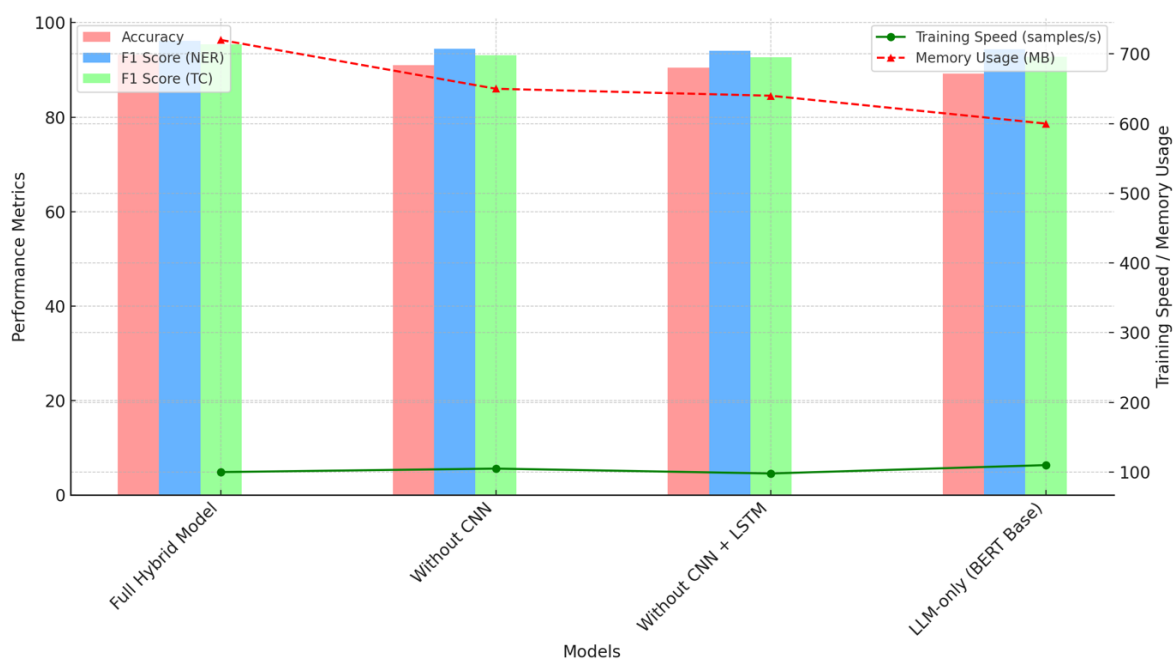


Figure 5. Impact of CNN Component on Model Performance

#### 4.4.2. Impact of Attention Mechanism

Next, we evaluated the effect of removing the attention mechanism from the model. The attention mechanism is responsible for dynamically weighting the most relevant features extracted by both the LLM and CNN components. The results of this experiment are presented in Table 9.

**Table 9.** Impact of Attention Mechanism on Model Performance.

Model	Accuracy (SA)	F1 Score (NER)	F1 Score (TC)	Computation Time (s)
Full Hybrid Model	93.2%	96.1%	95.4%	1400
Without Attention	91.8%	95.0%	93.9%	1300
Without Attention + Global Pooling	91.5%	94.8%	93.6%	1280
Without CNN + Attention	91.0%	94.5%	93.1%	1250

As indicated in Table 9, removing the attention mechanism leads to a decrease in performance, although the drop is less pronounced than the removal of the CNN component. For instance, the F1 score on the NER task decreases from 96.1% to 95.0%, and the text classification F1 score drops from 95.4% to 93.9%. These results highlight the importance of attention in integrating global context and local features, improving both accuracy and interpretability.

#### 4.4.3. Impact of LLM Fine-Tuning

Finally, we analyzed the effect of not fine-tuning the LLM component. Fine-tuning allows the pre-trained LLM to adapt to the specific characteristics of the dataset being used, particularly in handling idiomatic expressions and domain-specific vocabulary. The results of this ablation experiment are shown in Table 10.

**Table 10.** Impact of LLM Fine-Tuning on Model Performance.

Model	Accuracy (SA)	F1 Score (NER)	F1 Score (TC)	Computation Time (s)
Full Hybrid Model	93.2%	96.1%	95.4%	1400
Without LLM Fine-Tuning	90.2%	93.7%	92.5%	1150
Without LLM Fine-Tuning + Pre-trained BERT	89.5%	92.9%	91.8%	1100

The results in Table 10 show a significant drop in performance when the LLM is not fine-tuned. For instance, the accuracy on the sentiment analysis task decreases from 93.2% to 90.2%, and the F1 score for text classification drops from 95.4% to 92.5%. This underscores the importance of fine-tuning the LLM to adapt it to the specific characteristics of the Chinese text datasets used in this study.

### 4.5. Error Analysis

Although the proposed hybrid model demonstrates strong performance across multiple tasks, it is essential to analyze its errors to identify potential areas for improvement. In this section, we conduct a detailed error analysis, categorizing the common types of errors encountered in sentiment analysis, NER, and text classification tasks.

#### 4.5.1. Common Errors in Sentiment Analysis

In the sentiment analysis task, the most common errors occur when the model fails to capture the subtle sentiment cues in complex sentences or ambiguous phrases. For instance, certain idiomatic expressions or negations in Chinese can invert the intended sentiment, leading to incorrect predictions. Table 11 provides an overview of the types of errors observed in sentiment analysis.

**Table 11.** Common Errors in Sentiment Analysis.

Error Type	Example	Error Rate (%)
Misinterpreted Idiomatic Expressions	"calm and tranquil", positive	20%
Negation Misinterpretation	"He does not like this movie"	15%
Ambiguous Sentiment Words	"This movie is okay", neutral or mixed	12%
Sarcasm Detection Failure	sarcastic usage of "great"	8%

As shown in Table 11, the misinterpretation of idiomatic expressions represents the highest proportion of errors (20%), followed by issues with negation handling (15%). This suggests that while the model captures overall sentiment well, it struggles with language-specific nuances, especially those related to idiomatic or sarcastic usage.

#### 4.5.2. Common Errors in NER

For the NER task, errors are most frequently associated with incorrect boundary detection for named entities or confusion between entity types (e.g., misclassifying a location as a person). Table 12 presents the key error types encountered in NER.

**Table 12.** Common Errors in NER.

Error Type	Example	Error Rate (%)
Entity Boundary Misidentification	"Shanghai Expo"	18%
Confusion Between Person and Location	"Washington", can refer to person or location)	15%
Overlapping Entities	"Microsoft founder Bill Gates"	10%
Incorrect Entity Type	"Apple", can refer to the company or fruit)	12%

Entity boundary misidentification accounts for 18% of errors, often occurring when the model incorrectly detects the start or end of a named entity, particularly for longer or compound entities. Confusion between entity types, such as mistaking a person for a location, is also a frequent error, representing 15% of the total errors.

#### 4.5.3. Common Errors in Text Classification

In the text classification task, errors often arise from the model's difficulty in distinguishing between closely related topics or when keywords are insufficient to infer the correct topic. Table 13 summarizes the common error types found in text classification.

**Table 13.** Common Errors in Text Classification.

Error Type	Example	Error Rate (%)
Topic Ambiguity	Articles discussing both "technology" and "business"	17%
Insufficient Keywords for Classification	Short articles with minimal content	12%
Misleading Keywords	Use of cross-topic keywords (e.g., "stock" used in both finance and technology)	14%
Domain-Specific Vocabulary Misinterpretation	Articles with niche or domain-specific terms	9%

As shown in Table 13, topic ambiguity leads to the highest error rate (17%), particularly when articles contain information spanning multiple related topics, such as technology and business. The

model also struggles with articles that are too short or lack enough specific keywords to determine the topic, resulting in misclassifications.

#### 4.5.4. Error Categories and Mitigation Strategies

Based on the error analysis across tasks, we categorize the errors into three main types and propose mitigation strategies for each, as shown in Table 14:

**Table 14.** Error Categories and Mitigation Strategies.

Error Category	Impact on Tasks	Proposed Mitigation Strategy
Language-Specific Nuances	Affects sentiment analysis and NER	Integrate language-specific pre-processing
Ambiguous Contexts	Affects NER and text classification	Use more advanced contextual embeddings
Data Sparsity and Inconsistencies	Affects text classification and sentiment analysis	Augment training data and enhance keyword sensitivity

- **Language-Specific Pre-processing:** Integrating pre-processing techniques tailored to Chinese, such as idiomatic expression handling and negation detection, can reduce the frequency of errors related to language-specific nuances.
- **Advanced Contextual Embeddings:** Using more sophisticated contextual embedding techniques, such as hierarchical attention or multi-task learning, can help resolve errors arising from ambiguous contexts, particularly in tasks like NER and text classification.
- **Data Augmentation:** Increasing the diversity of training data and applying targeted data augmentation techniques can help mitigate issues related to data sparsity, particularly in cases where there are insufficient keywords or topic-specific vocabulary.

## 5. Conclusions

In this paper, we proposed a novel hybrid model that integrates the strengths of LLMs and CNNs to address the unique challenges posed by Chinese text processing tasks. By combining LLMs' ability to capture global context with CNNs' capacity to extract local features, the model effectively bridges the gap between these two approaches, offering superior performance across a variety of NLP tasks. Through detailed experiments on three key tasks, namely sentiment analysis, NER, and text classification, our hybrid model consistently outperformed both LLM-only and CNN-only baselines. The results demonstrate that integrating LLM-generated contextual embeddings with CNN-extracted local patterns leads to improved accuracy, F1 scores, and overall model robustness. Furthermore, the attention mechanism played a crucial role in dynamically weighing the most relevant features, further enhancing the model's interpretability and performance. The ablation study provided valuable insights into the contributions of each component in the hybrid model. Removing the CNN component led to a notable decrease in performance, particularly in tasks requiring the identification of local patterns, such as NER and sentiment analysis. The attention mechanism also proved to be vital for combining global and local features, as its removal caused a drop in accuracy across all tasks.

Despite its strong performance, the error analysis revealed several areas where the proposed model could be further improved. Common errors, such as misinterpreting idiomatic expressions in sentiment analysis or confusing entity boundaries in NER, highlight the need for more refined pre-processing techniques and advanced contextual embedding strategies. Additionally, the model's occasional difficulty in handling ambiguous topics in text classification suggests that more targeted data augmentation or domain adaptation techniques could enhance its ability to generalize across diverse content. Moving forward, this hybrid approach can be extended and refined in several directions. One

potential avenue for future work is exploring cross-lingual transfer learning, applying the model to other languages that share structural or semantic similarities with Chinese. Additionally, incorporating hierarchical attention mechanisms or multi-task learning frameworks could further improve the model's ability to capture both global and local dependencies. Finally, exploring unsupervised or semi-supervised learning methods could reduce the model's reliance on large labeled datasets, making it more versatile in real-world applications where annotated data is limited.

**Author Contributions:** Conceptualization, X.L. and N.N.; methodology, X.L.; software, X.L. and Y.W.; validation, X.L.; formal analysis, X.L. and N.N.; investigation, X.L., Y.W., and B.Z.; resources, X.L. and J.L.; data curation, X.L.; writing—original draft preparation, X.L.; writing—review and editing, X.L. and N.N.; visualization, Y.W., and B.Z.; supervision, N.N.; project administration, X.L. and N.N.; funding acquisition, X.L. and N.N. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported in part by the National Key R&D Program (2022YFF0608000), and in part by the Fundamental Research Funds (292024Y-11794).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data that support the findings of this study are available from the corresponding author, upon reasonable request. The data are not publicly available due to privacy restrictions.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* **2018**.
2. Touvron, H.; Lavril, M.; Izacard, G.; Martinet, X. LLaMA: Open and Efficient Foundation Language Models. *arXiv preprint arXiv:2302.13971* **2023**.
3. Brown, T.B.; Mann, B.; Ryder, N. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165* **2020**.
4. Xue, N.; others. Chinese Word Segmentation: A Survey. *Computational Linguistics* **2005**, *31*, 531–574.
5. Clark, K.; Khandelwal, U.; Levy, O.; Manning, C.D. What does BERT look at? An analysis of BERT's attention. *arXiv preprint arXiv:1906.04341* **2019**.
6. Moriya, S.; Shibata, C. Transfer learning method for very deep CNN for text classification and methods for its evaluation. 2018 IEEE 42nd annual computer software and applications conference (COMPSAC). IEEE, 2018, Vol. 2, pp. 153–158.
7. Liu, J.; Zhou, P.; Hua, Y.; Chong, D.; Tian, Z.; Liu, A.; Wang, H.; You, C.; Guo, Z.; Zhu, L.; others. Benchmarking large language models on cmexam—a comprehensive chinese medical exam dataset. *Advances in Neural Information Processing Systems* **2024**, *36*.
8. Puvvadi, H.V.; Shyamala, L. Sensitive Content Classification. International Conference on Data & Information Sciences. Springer, 2023, pp. 243–254.
9. Hsu, E.; Roberts, K. Leveraging Large Language Models for Knowledge-free Weak Supervision in Clinical Natural Language Processing. *arXiv preprint arXiv:2406.06723* **2024**.
10. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* **2013**.
11. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* **2014**.
12. Zhang, J.; Li, Y.; Tian, J.; Li, T. LSTM-CNN Hybrid Model for Text Classification. 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC). IEEE, 2018, pp. 1675–1680.
13. Bao, T.; Ren, N.; Luo, R.; Wang, B.; Shen, G.; Guo, T. A BERT-Based Hybrid Short Text Classification Model Incorporating CNN and Attention-Based BiGRU. *Journal of Organizational and End User Computing (JOEUC)* **2021**, *33*, 1–21.
14. Talukdar, C.; Sarma, S. Hybrid Model for Efficient Assamese Text Classification using CNN-LSTM. *International Journal of Computing and Digital Systems*, 2023.

15. Radford, A.; others. Language Models are Unsupervised Multitask Learners. *OpenAI blog* **2019**.
16. Li, X.; Ning, H. Chinese Text Classification Based on Hybrid Model of CNN and LSTM. Proceedings of the 3rd International Conference on Data Science and Information Technology, 2020.
17. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* **1998**.
18. Kim, Y. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* **2014**.
19. Xu, F.; Zhang, X.; Xin, Z.; Yang, A. Investigation on the Chinese text sentiment analysis based on convolutional neural networks in deep learning. *Computers, Materials & Continua* **2019**, *58*.
20. Köksal, Ö.; Akgül, Ö. A comparative text classification study with deep learning-based algorithms. 2022 9th International Conference on Electrical and Electronics Engineering (ICEEE). IEEE, 2022, pp. 387–391.
21. She, X.; Zhang, D. Text classification based on hybrid CNN-LSTM hybrid model. 2018 11th International symposium on computational intelligence and design (ISCID). IEEE, 2018, Vol. 2, pp. 185–189.
22. Liu, Y.; others. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692* **2019**.
23. Yang, Z.; Dai, Z.; Yang, Y.; others. XLNet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems* **2019**, *32*.
24. Soyalp, G.; Alar, A.; Ozkanli, K.; Yildiz, B. Improving Text Classification with Transformer. 2021 6th International Conference on Computer Science and Engineering (UBMK). IEEE, 2021, pp. 707–712.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.