International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 13 Issue 7 ǁ July 2025 ǁ PP. 100-106

Performance Comparison of CNN and BiLSTM with FastText Word Embeddings for Chinese Weibo Sentiment Analysis

Meiqin Wu[1], Zhan Wen[1,2*], Peng Xiao[1], Wenzao Li[1,2]
1 School of Communication Engineering, Chengdu University of Information Technology, Chengdu, 610225, Sichuan, China .*
2 Meteorological information and Signal Processing Key Laboratory of Sichuan Higher Education Institutes of Chengdu University of Information Technology, Chengdu, 610225, Sichuan, China .
**Corresponding author(s). E-mail(s): wenzhan@cuit.edu.cn*

Abstract
With the rapid development of Chinese social media, Weibo, as a mainstream platform, has generated a large amount of text data with emotional tendencies. The demand for applying this data in fields such as public opinion monitoring and user feedback analysis is becoming increasingly urgent. However, the complexity of emotional expression and the dynamic changes in vocabulary in Chinese Weibo texts have placed higher demands on sentiment analysis models. This paper addresses the sentiment analysis task for Chinese Weibo text, comparing the performance differences between two classic deep learning models: Convolutional Neural Networks (CNN) and Bidirectional Long Short-Term Memory Networks (BiLSTM). A single-modal comparison model was constructed using publicly available FastText pre-trained word vectors and the Weibo_Senti_100k dataset, with data cleaning and tokenization applied as preprocessing steps. The experimental results show that FastText word vectors provide effective semantic representations for the model. Combined with the local feature extraction capabilities of CNN, it performs better in sentiment analysis tasks, with an accuracy rate, recall rate, and F1 score of 97.93%, significantly higher than the 95.99% achieved by BiLSTM.Additionally, its average training time per epoch is only 28% of that of BiLSTM. This difference stems from the CNN's efficiency in locally extracting high-frequency sentiment keywords from Weibo text. This study provides empirical evidence for model selection in Chinese Weibo sentiment analysis, confirming the efficiency and advantages of the "FastText+CNN" framework.

Keywords: *FastText; Chinese Weibo text sentiment analysis; CNN; BiLSTM*

Date of Submission: 09-07-2025 Date of acceptance: 23-07-2025

I. INTRODUCTION

With the continuous development of information technology and social networks, coupled with the explosive growth of online text information, text sentiment analysis technology plays a crucial role in online public opinion monitoring and product development analysis. As one of China's most representative social platforms, Weibo witnesses Chinese netizens commenting on various trending topics daily, generating massive textual data. By conducting fine-grained sentiment analysis on this textual data, we can uncover people's perspectives, which hold significant theoretical and practical value.

Sentiment analysis, also known as opinion mining, is one of the core tasks in the field of natural language processing (NLP), focusing on identifying and extracting subjective information from text data. The primary objective of sentiment analysis is to determine the emotional tone conveyed by the text—whether it is positive, negative, or neutral [1]. Chinese text sentiment analysis, as a key branch of this field, aims to accurately identify and deeply analyze the emotional tendencies embedded in text.

Notably, Chinese social media text exhibits distinct linguistic features, such as frequent internet neologisms and fragmented sentence structures, which pose new challenges and opportunities for sentiment analysis. Traditional word embedding models (such as Word2Vec) have significant limitations when handling out-of-vocabulary (OOV) words—their independent word vector representations cannot cover the dynamic neologisms found in social media. In contrast, FastText enhances adaptability to dynamic changes in Chinese vocabulary through character-level n-gram subword modeling, making it more suitable for social media text scenarios.

In deep learning models, CNN and BiLSTM networks demonstrate different advantages in sentiment analysis: CNN efficiently captures local sentiment word clusters through convolution operations, making it suitable for short text scenarios like Weibo; while BiLSTM captures long-distance semantic dependencies such as "Although the taste is average, the cost-effectiveness is high" through bidirectional temporal modeling.

Based on the above background, this study proposes to integrate the pre-trained Chinese word vectors of FastText with deep learning architectures. The research motivation is to verify the compatibility of FastText pre-trained word vectors with two deep learning architectures, and to systematically compare the performance. differences of CNN and BiLSTM in Chinese microblog sentiment analysis. The research contributions include:

(1) Introducing FastText pre-trained word vectors and verifying their compatibility with CNN and BiLSTM in Chinese Weibo sentiment analysis.

(2) Conducting comparative experiments on the public dataset Weibo_Senti_100k to provide empirical evidence for model selection.

II. RELATED WORKS

Early Chinese sentiment analysis primarily relied on traditional models such as Support Vector Machines (SVM) and Naive Bayes. These traditional machine learning algorithms have low computational complexity and strong interpretability. They demonstrated excellent performance on small datasets and were commonly used sentiment analysis methods before the rise of deep learning [2]. Therefore, some researchers at the time utilized traditional machine learning algorithms to construct classification models. Chikersal et al. [3] developed a support vector machine text classification scheme integrated with semantic rules, which demonstrated good performance in text sentiment analysis. Reference [4] integrated the Naive Bayes algorithm with genetic algorithms to design an innovative text classification method, which significantly improved classification accuracy in movie review analysis.

When handling high-noise, unstructured text in scenarios such as social media, the manual feature extraction capabilities of traditional models are limited and unable to automatically capture semantic associations and contextual dependencies in text, leading to insufficient generalization capabilities in complex scenarios. Deep neural networks can automatically learn text features. Among these, CNNs excel at capturing local n-gram features. Reference [5] proposed a CAT-FWE model based on channel attention TextCNN and feature word extraction for binary and fine-grained sentiment classification tasks on Chinese short texts. Recurrent neural networks (such as LSTM and BiLSTM) can remember long-distance dependency information in text through memory units. Lai et al. [6] successfully classified the sentiment of short texts by using BiLSTM and graph attention networks to extract contextual information and sentence syntactic information, respectively. CNNs are more efficient at local feature extraction, while BiLSTM and other models are better suited for processing semantically complex long texts, forming a complementary technical approach in different task scenarios.

In terms of word vector representation, early studies mostly used randomly initialized word embedding vectors, with models gradually learning semantic relationships between words during training. With the development of pre-training techniques, static word vectors trained on large-scale corpora (such as Word2Vec and GloVe) have been widely applied in Chinese sentiment analysis. P. Wang et al. [7] used Word2Vec to train a word embedding model, extract key sentences, and combine grammatical rules with word vector similarity analysis to determine the sentiment orientation of documents. Multiple experiments demonstrated that this method effectively improved the accuracy and adaptability of sentiment analysis. However, traditional Word2Vec has limited adaptability to dynamic vocabulary; in contrast, FastText enhances adaptability to dynamic changes in Chinese vocabulary via subword-level modeling. S.Sadiqd et al. [8] combined CNN with FastText embeddings, outperforming other models in accurately distinguishing machine-generated text in dynamic social media environments. Shumaly et al. [9] used the FastText method to create word embeddings and compared the results of CNN, BiLSTM, Logistic Regression, and Naïve Bayes models. The results showed that FastText and CNN performed best, achieving an AUC of 0.996 and an F-score of 0.956. In summary, although existing research has made significant progress in optimizing sentiment analysis models and evolving word vector technologies, there remains a research gap in the systematic performance comparison between CNN and BiLSTM in the context of Chinese Weibo text scenarios, particularly lacking a systematic performance comparison between the combination of FastText word vectors with CNN and BiLSTM in the scenario of Chinese Weibo short texts.

III. Model Structure

Two model architectures are designed in this study, namely:

FastText + CNN model: Weibo text is tokenized and input into the CNN model via FastText vector representation, and multiple convolutional kernels are used to extract the sentiment features and classify them through fully connected layers.

FastText + BiLSTM model: the FastText vectors are used as inputs to the BiLSTM network to obtain the semantic representations of the front and back texts, and then classified by the model through the fully connected layer.

3.1 FastText Model

The model framework of FastText shares significant similarities with the CBOW architecture of Word2Vec, both employing a three-layer structure: input layer, hidden layer, and output layer. However, they exhibit distinct differences in design objectives and feature processing: CBOW learns word embeddings by predicting central words based on contextual word vectors, while FastText averages the bag-of-words features and n-gram substring feature vectors of the text as input, mapping them through a single-layer neural network to a softmax output layer with the number of categories (supporting hierarchical optimization). This mechanism enhances adaptability to dynamic changes in Chinese vocabulary through subword-level modeling and achieves training speeds 3-5 times faster than traditional models, making it more suitable for rapid classification of large-scale text.

The Chinese word vectors used in this paper are sourced from the officially released FastText pre-trained model by Facebook AI Research. This model is trained on a large-scale Chinese corpus, with word vector dimensions of 300, and possesses subword modeling capabilities, making it suitable for the semantic representation of social media text. The FastText model framework is illustrated in the figure below:

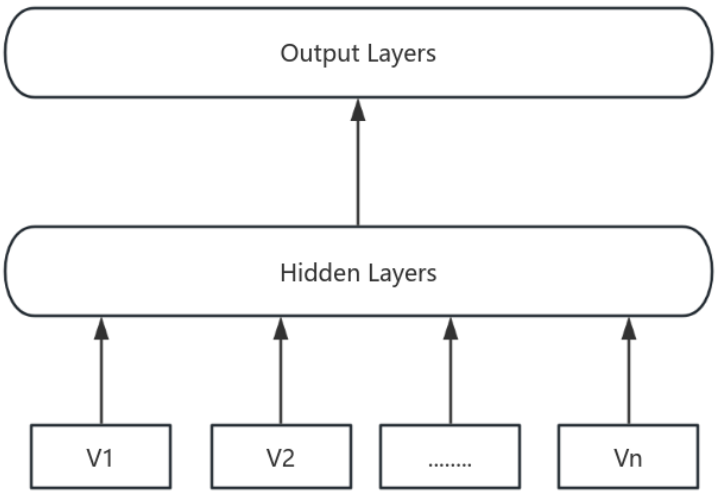

Figure 1. FastText Model Framework

3.2 CNN and BiLSTM Model Architectures

To ensure fairness in the comparison experiment, CNN and BiLSTM use the same input layer and embedding layer. The input layer receives a sequence of word indices after word segmentation, and the embedding layer loads 300-dimensional FastText pre-trained Chinese word vectors to construct a fine-tunable embedding matrix, mapping discrete indices to continuous semantic representation vectors, thereby enhancing the model's initial understanding of word semantics.

3.2.1 CNN Model

As an important architecture in the field of deep learning, CNN was initially used for image processing and later adapted to the field of NLP. CNN captures local features of text through convolution operations, and its multi-scale convolution kernel design (window sizes 3, 4, and 5) is suitable for extracting key semantic features from short texts. The model captures n-gram features through convolution layers and combines them with maximum pooling layers for dimension reduction, thereby retaining the "local perception" characteristics while adapting to the sequential structure of natural language.

In this study, the CNN model sequentially extracts local phrase-level sentiment features through multi-scale convolutional operations (convolution kernel window sizes of 3, 4, and 5), extracts primary features via max pooling, applies Dropout regularization to suppress overfitting, and finally outputs sentiment classification results through a fully connected layer.

3.2.2 BiLSTM Model

Compared to CNNs, recurrent neural networks (RNNs) are more suitable for processing sequence data with contextual dependencies. LSTM networks address the gradient vanishing problem in standard RNN models during long sequence learning by introducing forget gates and memory mechanisms. BiLSTM further

enhances contextual modeling capabilities by propagating information in both forward and backward directions across a sequence, thereby more comprehensively capturing semantic dependencies in text.

In this study, the BiLSTM model employs a 3-layer bidirectional LSTM, with each layer containing 128 hidden units. The forward LSTM captures the preceding semantic context, while the backward LSTM captures the subsequent semantic context. The bidirectional outputs are concatenated to obtain a complete contextual representation. Similarly, a fully connected layer is connected after the Dropout layer for classification prediction.

3.3 Model Training Design

The overall flowchart of the model is as follows:

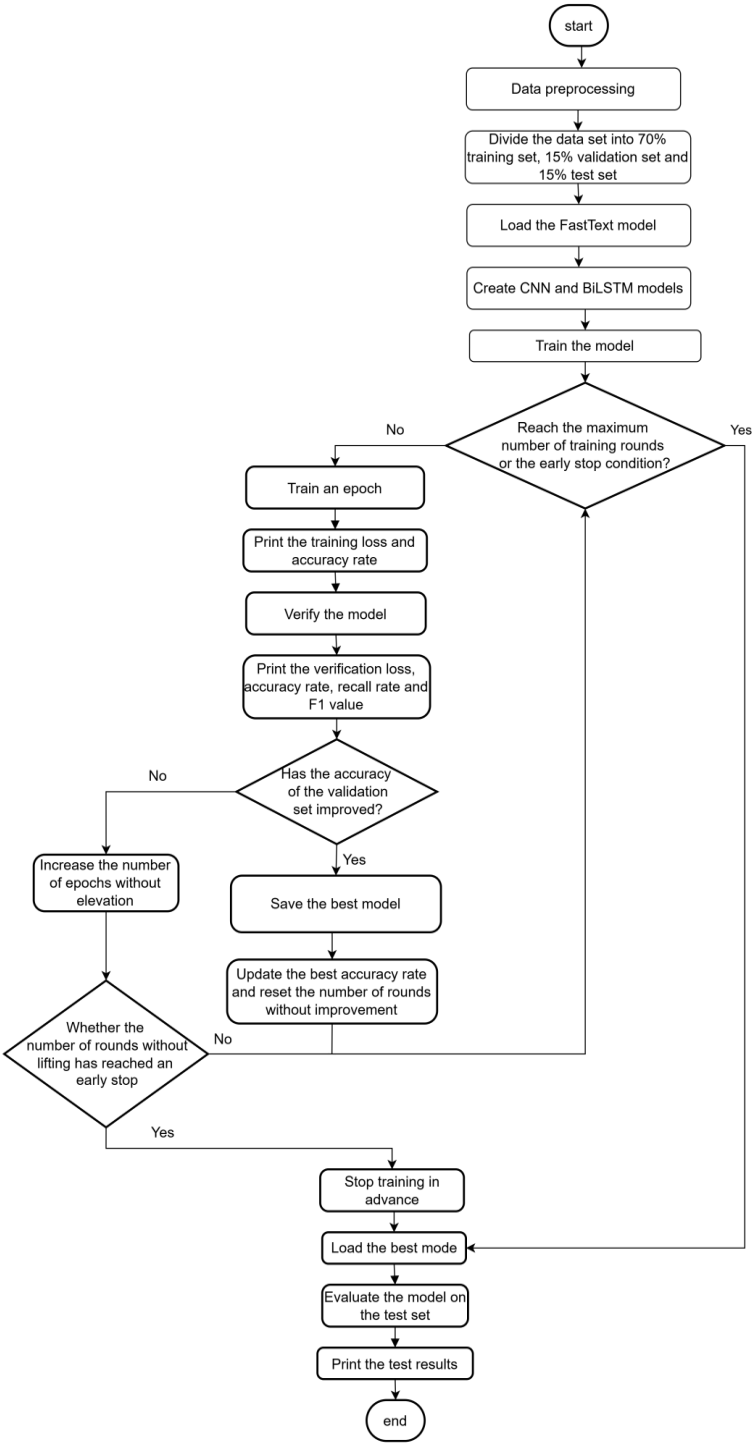

Figure 2. Overall design flowchart

The first step is data preprocessing, dividing the dataset, loading Chinese FastText pre-trained word vectors, and constructing fine-tunable embedding matrices to deal with unregistered words; designing the CNN architecture and BiLSTM architecture, respectively, and adopting the Adam optimizer in the training phase combined with the early-stopping mechanism (terminate if the loss does not decrease in 5 consecutive rounds of validation); and finally evaluating the performance of the model on the test set through the accuracy, recall, and F1 value.

IV. EXPERIMENTS and DISCUSSION

4.1 dataset

In this paper, we use the Weibo_Senti_100k public dataset, which contains more than 100,000 Sina Weibo texts with sentiment annotations, among which positive and negative sentiment comments account for roughly half of the total, about 50,000 comments each. The comments are labeled with "0" and "1", where "0" denotes "negative" and "1" denotes "positive". We randomly divide the dataset into a 70% training set, a 15% validation set, and a 15% testing set. The distribution of the dataset is as follows:

Table 1. Dataset distribution

Category	1	0	Total
Training set	41839	42152	83991
Validation set	9054	8944	17998
Test Set	9100	8899	17999

Data preprocessing: In natural language processing tasks, dataset preprocessing is a crucial step that directly affects the performance of the model and the training efficiency. In the data preprocessing stage, we first use regular expressions to clean the text, remove HTML tags, hyperlinks, and other special characters, and normalize them; we use Jieba for word segmentation and filter stop words; we then cache and load the FastText model and construct the vocabulary list with special tokens and embedding matrices; finally, we convert the text into indexes and create the data loader. These steps convert the original Chinese text into a format suitable for the model, which lays the foundation for the training of the sentiment analysis model.

4.2 Experimental Parameter Settings

The Adam optimizer was selected, with a learning rate of 0.001 and 20 epochs. This configuration is based on the standard settings in the field of Chinese sentiment analysis. To balance computational resource efficiency and gradient stability, the batch size was set to 128, and an early stopping strategy was implemented to prevent model overfitting. All models were operated in the same software environment. The hyperparameter settings for the experiment are shown in Table 2.

Table 2.Experimental parameters

Parameters	Parameter Value
Batch _size	128
epochs	20
Early_ stop_ patience	5
optimizer	Adam
learning rate	0.001

4.3 Experimental results and analysis

After the experiment, the experimental results are shown in Table 3:

Table 3. Experimental results

	Loss	Accuracy	Recall rate	F1	Average training time per epoch
CNN	0.0628	97.931%	97.931%	97.932%	8min12s
BiLSTM	0.1085	95.993%	95.993%	95.993%	29min5s

The experimental results illustrate that the CNN model outperforms BiLSTM in all the metrics. The CNN model achieves 97.931% in accuracy and recall, and 97.932% in F1. While the BiLSTM model is 95.993%. In addition, the loss value of CNN is 0.0628, which is significantly lower than that of BiLSTM (0.1085), which is different from the traditional conclusion that "BiLSTM is more suitable to deal with semantic dependencies". There are several reasons for this difference:

(1) Text feature variability: Most of the samples in the Weibo_Senti_100k dataset are about 78 words in length, which is a typical short text scenario. The sentiment polarity of this type of text is highly dependent on local high-frequency word clusters, and CNNs are naturally good at extracting significant sentiment features (e.g., emotional keywords and phrases) from local regions, and thus can capture the key information in short texts more efficiently. On the other hand, long-distance semantic dependencies (e.g., "although... but..." contrastive relationship), which BiLSTM is good at, appear less frequently in short texts, making it difficult to take advantage of its bidirectional temporal modeling.

(2) Model structure fitness: BiLSTM performs better in modeling contextual relationships of long texts, but is prone to redundant modeling or noise propagation for short texts, thus affecting classification performance. In contrast, the parallel convolutional operation of CNN can simultaneously extract multi-scale local features, and retain significant sentiment features through the maximum pooling layer, with high computational efficiency and clear goals.

(3) Task-specific adaptability: This experiment is a binary classification task, where sentiment polarity determination relies heavily on explicit keywords (e.g., "good" or "bad") rather than complex semantic reasoning. The "keyword capture" feature of CNNs is more efficient for such tasks. While BiLSTMs excel at handling long-range dependencies (e.g., complex sentence structures), this advantage does not translate into performance improvements in simple binary classification tasks and instead increases computational overhead.

Additionally, there is a significant difference in runtime between the two models: the average training time per epoch for CNN is 492 seconds, while the average training time per epoch for BiLSTM is 1,745 seconds, making the average training time per epoch 3.55 times longer for BiLSTM. This difference stems from the computational characteristics of the model structures: CNN's convolution operations have inherent parallelism (multi-scale convolution kernels can simultaneously process different local features), while BiLSTM's sequential computations must be executed in sequence, resulting in lower efficiency under the same hardware conditions. For short text analysis scenarios such as microblogs and comments that require rapid iteration, the time efficiency advantage of CNN makes it more suitable for deployment in resource-constrained environments, further validating the practicality of the "FastText+CNN" framework.

V. CONCLUSION

This study focuses on Chinese social media scenarios and explores sentiment analysis methods that combine FastText with deep learning models, comparing the applicability of two architectures: CNN and BiLSTM. The results indicate that CNN is better suited to the characteristics of short Weibo texts, which are dense with local sentiment features. Overall, the sentiment recognition framework based on "FastText + CNN" offers superior efficiency and generalization capabilities, providing a practical foundation for applications such as public opinion monitoring and user sentiment feedback analysis. This study confirms that model selection must be closely aligned with data characteristics: for sentiment analysis of short texts such as Weibo posts and comments, the CNN+FastText combination is more practical; however, when tasks involve long texts (e.g., news comments, novels) or complex semantic reasoning, the advantages of BiLSTM become more evident. Future research could explore combining the strengths of both approaches (e.g., using CNN to extract local features followed by BiLSTM modeling) or introducing attention mechanisms to enhance key information capture, thereby further improving model performance. Additionally, exploring multi-modal fusion, fine-grained sentiment classification, and lightweight deployment could enhance the model's application breadth and practical usability.

ACKNOWLEDGEMENT

This research was supported by the Sichuan Science and Technology Program, Soft Science Project (No.2022JDR0076) and Sichuan Province Philosophy and Social Science Research Project(No. SC23TJ006). We also would like to thank the sponsors of Meteorological Information and Signal Processing Key Laboratory of Sichuan Higher Education Institutes of Chengdu University of Information Technology.

REFERENCES

[1]. Albladi, A. , Islam, M. , & Seals, C..(2025). Sentiment analysis of twitter data using nlp models: a comprehensive review. IEEE Access, 13.

[2]. Li, Y., Zhou, B., Niu, Y., & Zhao, Y. (2025). Fine grained sentiment analysis on microblogs based on graph convolution and self attention graph pooling. Applied Intelligence, 55(2), 92.

[3]. Chikersal, P. , Poria, S. , & Cambria, E. . (2015). SeNTU: Sentiment Analysis of Tweets by Combining a Rule-based Classifier with Supervised Learning. International Workshop on Semantic Evaluation.

[4]. Govindarajan, M. (2013). Sentiment analysis of movie reviews using hybrid method of naive bayes and genetic algorithm. International Journal of Advanced Computer Research, 3(4), 139.

[5]. Liu, J., Yan, Z., Chen, S., Sun, X., & Luo, B. (2023). Channel attention TextCNN with feature word extraction for Chinese sentiment analysis. ACM Transactions on Asian and Low-Resource Language Information Processing, 22(4), 1-23.

[6]. Lai, Y., Zhang, L., Han, D., Zhou, R., & Wang, G. (2020). Fine-grained emotion classification of Chinese microblogs based on graph convolution networks. World Wide Web, 23, 2771-2787.
[7]. Wang, P. , Luo, Y. , Chen, Z. , He, L. , & Zhang, Z. . (2019). Orientation analysis for chinese news based on word embedding and syntax rules. IEEE Access, PP(99), 1-1.
[8]. Sadiq, S., Aljrees, T., & Ullah, S. (2023). Deepfake detection on social media: leveraging deep learning and fasttext embeddings for identifying machine-generated tweets. IEEE Access, 11, 95008-95021.
[9]. Shumaly, S., Yazdinejad, M., & Guo, Y. (2021). Persian sentiment analysis of an online store independent of pre-processing using convolutional neural network with fastText embeddings. PeerJ Computer Science, 7, e422.