Creativity and Innovation
2025, VOL. 9, NO. 5, 248-253
DOI: 10.47297/wspciWSP2516-252750.20250905

Sentiment Analysis of Product Reviews Based on the RoBERTa-CNN-BiLSTM-Attention

Chenyi Xiao

Harbin Institute of Technology,Weihai, Shandong, 264200, China

ABSTRACT
Automated sentiment analysis has become essential for extracting subjective opinions from text due to the rapid expansion of online text data. The RoBERTa-CNN-BiLSTM-Attention model is a hybrid architecture that combines deep learning elements with pre-trained language models to address the shortcomings of conventional approaches in semantic ambiguity handling and multi-scale feature capture. This model uses CNN to extract local sentiment phrase features, BiLSTM to simulate bidirectional sequence dependencies in lengthy texts, RoBERTa for deep semantic encoding, and an Attention method to highlight important sentiment information. By doing this, "local features + global logic + critical information" can be collaboratively modeled. Test accuracy on Amazon's 400,000-item review dataset is 95.49%, which is a considerable improvement above single-component models and baseline models like BERT and Word2Vec. These outcomes provide a practical solution for complicated text sentiment analysis and confirm the efficacy of multi-component fusion structures.

KEYWORDS
Sentiment analysis; RoBERTa-CNN-BiLSTM-Attention; Product reviews; Hybrid architecture

1 Introduction

Human-generated text data has increased dramatically with the growth of the internet. Examples include product reviews on e-commerce sites and social media postings and comments, reader comments on news stories and forum debates, internal business user feedback, and customer support conversations. Numerous subjective viewpoints and emotional expressions can be found in these books. The ability to quickly extract "sentiment signals" from large amounts of text is crucial for businesses, governments, and individuals to use in decision-making. Traditional manual processing of this kind of data, however, is incredibly inefficient. In addition to being laborious and time-consuming, manually sorting through each piece to identify sentiment is unable to handle the real-time processing demands of large amounts of data. As a result, there is a growing need for "automated sentiment interpretation" [1].

A crucial area of Natural Language Processing (NLP), sentiment analysis uses computational methods to automatically detect, extract, and measure subjective emotions—such as positive, negative, or more nuanced sentiments—from speech, text, and other data [2]. Extracting "subjective attitudes" from enormous volumes of unstructured text was the main goal in its early phases. The term "sentiment analysis" was not yet widely accepted at this time and was more frequently used as "opinion mining" [3]. To determine scores based on sentiment words for analysis, it used manually created sentiment dictionaries and basic syntactic principles, such as dictionaries of negative words, adverbs of degree, and stop words [4]. Nevertheless, this manual method was limited to simple sentences and suffered from ambiguity and context dependency [5]. Later, as machine learning technologies advanced, sentiment analysis gained additional tools and was formally recognized as a separate field of study [6]. Sentiment analysis was turned into a classification challenge by researchers. Sentiment dictionaries and machine learning became the standard method when classifiers were trained using machine learning algorithms like Naive Bayes [7], Support Vector Machines (SVM) [8], and Maximum Entropy Models [9] on manually labeled data like "positive comments" and "negative comments" [10]. This method, however, depends on human feature engineering and calls for manually created characteristics such as "word frequency" and "number of sentiment words." On contextually complicated texts that incorporate metaphors or irony, it performs poorly. Instead of identifying sentiment toward particular elements, analysis granularity usually evaluates the sentiment of the text as a whole.

Deep learning technologies started to spread throughout the NLP field during the rapid development era [11]. Models such as "TextCNN" have become popular frameworks for sentiment analysis because Convolutional Neural Networks (CNNs) are excellent at capturing local sentiment features [12]. Because of their ability to manage sequential dependencies, recurrent neural networks (RNNs/LSTMs) are better suited for sentiment analysis in lengthy texts [13]. Bidirectional semantic dependencies are best captured by the bidirectional Long Short-Term Memory (BiLSTM), which combines forward and backward LSTMs [14]. "Attribute-level sentiment analysis" became a research priority as application demands changed; it involves first identifying key properties in text and then assessing sentiment toward those features. However,

* **Corresponding Author:** Chenyi Xiao, 2022210968@stu.hit.edu.cn

large volumes of manually annotated, high-quality data are needed for deep learning models, and data scarcity is especially noticeable in specialist fields or low-resource languages. Sentiment analysis techniques were drastically changed with the advent of attention-based models in 2017 [15] and the pre-trained language model BERT in 2018 [16], which also broadened the field's focus beyond "text" to "multimodal" data. RoBERTa is an improved version of BERT that has more training data, bigger batch sizes, and additional model parameters [17]. During this time, sentiment analysis became widely commercialized: e-commerce sites used it to evaluate user reviews and improve their products; social media used it to track public sentiment; and the financial industry used it to predict market volatility by analyzing news and forum sentiment [18]. At the same time, studies on hybrid architectures that combine various models have been developing, and a trend that has emerged is the integration of deep learning with pre-trained models [19-20].

This paper presents a RoBERTa-CNN-BiLSTM-Attention model that synthesizes the advantages and disadvantages of models put out across these eras. Using a hierarchical fusion architecture, RoBERTa provides a deep semantic foundation and addresses the problem of "semantic ambiguity" [21]. CNN enhances the capture of short-sequence sentiment features by extracting local sentiment phrases [22], BiLSTM integrates contextual sentiment logic in long texts by modeling bidirectional sequence dependencies [23], and the Attention mechanism highlights important information by removing redundant content [24] and focusing on core sentiment words. Complementary multi-scale features are made possible by this joint modeling of "local features + global logic + key information."

2 Model architecture

Input preprocessing and pretraining, feature extraction, sequence modeling, feature focusing, and fully linked feature output are the five primary layers that make up the RoBERTa-CNN-BiLSTM-Attention model. A thorough description of the working mechanisms for each of these five modeling elements may be found below.

2.1 The pre-training layer and input preprocessing

By using better pre-training techniques, RoBERTa, an optimized variant of BERT, dramatically improves performance. The NSP task is eliminated because its unclear goal could impede model learning. MLM is the exclusive focus of training in order to improve long-sequence processing. Furthermore, during training, masks are created at random. This method compels the model to acquire more robust semantic representations than BERT's fixed masks during preprocessing. Additionally, RoBERTa uses byte-level BPE tokenization, longer training times, bigger batch sizes, and larger datasets. Training stability is improved by modifying optimizer parameters and learning rate techniques.

RoBERTa is used in this study as the pretraining and input preprocessing tool. Because the model automatically learns semantic information from pretraining, it provides richer semantic representations than conventional word embeddings like Word2Vec and does away with the necessity for manually created sentiment dictionaries.

2.2 CNN local layer for feature extraction

Originally a feedforward neural network, convolutional neural networks (CNNs) are deep learning models designed specifically to interpret grid-structured input. Text can be thought of as a one-dimensional sequence in Natural Language Processing (NLP), where each word is transformed into a high-dimensional vector using RoBERTa, so that the text input format can be expressed as [sequence length, feature dimension]. In this situation, local sequence features can be extracted from text using CNN's convolution function.

CNN is used as a local feature extractor in this investigation. It conducts convolution computations for every kernel, eliminates superfluous dimensions, and converts RoBERTa's output to the convolution input format. The most noticeable characteristics that each kernel extracts are kept when the convolved features are downsampled using pooling. The CNN's final output, which serves as input for the next BiLSTM layer, is finally obtained by concatenating the feature outputs from various-sized kernels.

2.3 Layer for BiLSTM sequence modeling

An improved recurrent neural network called Long Short-Term Memory (LSTM) was created to overcome the drawbacks of conventional recurrent neural networks (RNNs). These drawbacks include the inability to record far-off information in lengthy sequences and the vulnerability to gradients that vanish or explode. By simultaneously using forward and backward LSTMs to record bidirectional sequence information, Bidirectional LSTM (BiLSTM) is an extension of LSTM. BiLSTM's bidirectional information fusion allows for more precise modeling of contextual settings at each sequence position, whereas regular LSTM simply records unidirectional temporal dependencies. Because of this, it is especially well-suited for tasks in natural language processing that call for complete context knowledge, including named entity recognition and sentiment analysis.

In this work, CNN-extracted features are processed using BiLSTM. Its bidirectional character improves feature representation through parameter learning by simulating bidirectional "pseudo-temporal" interactions among these local features. It improves model accuracy by highlighting additional elements that are essential for sentiment categorization

when paired with attention methods.

2.4 Layer of attention feature focus

Simulating the human cognitive process of "selective attention" is the fundamental function of the attention mechanism. Key information is given higher weights and secondary information is given lower weights by determining the "importance weights" of the various positions in a sequence. In the end, it uses weighted summing to produce more representative features. The classic version of this technique is Multi-Head Self-Attention. In order to capture correlations across several dimensions, it breaks down input data into numerous subspaces, with each brain independently calculating attention. To improve the model's capacity to represent intricate sequential interactions, the outputs from several heads are subsequently concatenated, combining multidimensional attention information. Sequence-to-vector conversion is the main focus of global attention pooling. It highlights the most crucial components for sentiment evaluation by performing weighted summation of sequence features after learning the important weights of each location for the final classification. Global Attention Pooling dynamically learns weights, allowing it to more flexibly adjust to a variety of sentiment expressions than standard pooling, which considers all positions identically.

These two attention systems cooperate within the model: By working with the sequence features that the BiLSTM produces, Multi-head Self-Attention improves the modeling ability for local emotion segments and increases the precision of the sequence features. By condensing the optimum sequence features into sentence-level representations, Global Attention Pooling concentrates on the most important global information for classification and serves as the foundation for the final sentiment assessment.

2.5 Feature output layer completely connected

The model uses a two-layer linear structure with ReLU activation to function as the classifier in the deep learning architecture. The expressive capability of this configuration is superior to that of single-layer linear classifiers, allowing for the learning of more intricate feature-category mappings. In order to successfully prevent overfitting and force the learning of more robust features, a Dropout layer is used. Semantic encoding → local characteristics → sequence dependencies → key focus → classification is the general logic of the entire framework. Deep semantic foundations are provided by RoBERTa, phrase-level sentiment is captured by CNN, sequence correlations are modeled by BiLSTM and Self-Attention, global attention pooling filters important information, and sentiment predictions are finally produced by the classification head. Together, these elements tackle fundamental issues in sentiment analysis, which qualifies the model for judging sentiment polarity in intricate texts such as product reviews. The figure shows the overall architecture of the model.

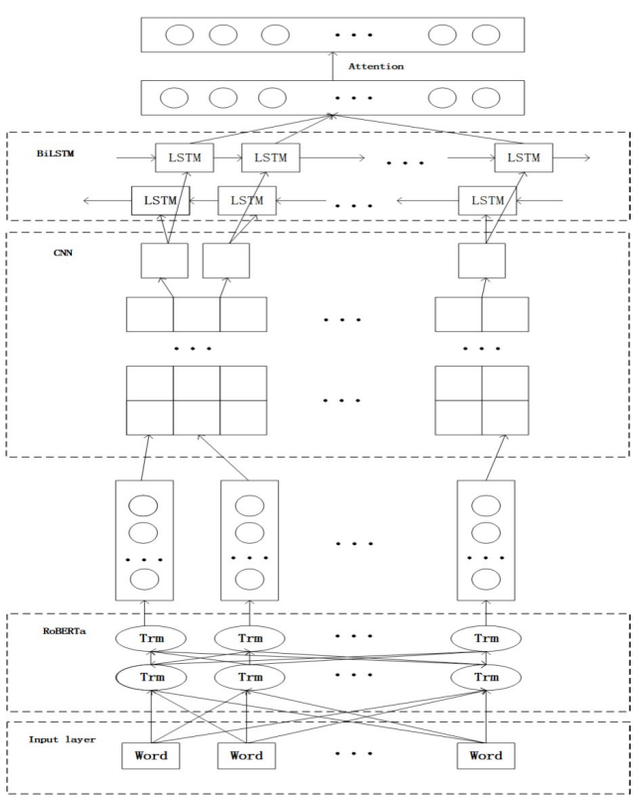

Figure 1

3 Experimental process and results

A total of 400,000 product reviews in TXT format were chosen for this experiment from Amazon's official e-commerce platform. Every data point's sentiment orientation was tagged for each text file. The original labels were transformed into binary classification task labels by the code, which separated the data into text and labels. Positive reviews were mapped to 1 and negative reviews to 0. The pre-trained BaseRoBERTa wordizer was used to encode the text, transforming natural English into tensors that could be processed by the model. After then, the dataset was divided in an 8:2 ratio between training and testing sets. Using Torch and its submodules for neural network creation and optimization, the model was constructed in a Conda environment. Convolutional layers, recurrent layers, attention layers, fully linked layers, regularization, and activation functions are all included in the core model architecture, which is based entirely on Torch's nn module and is wrapped in a sequential manner. Dataset and DataLoader deal with standards, while Tensor objects manage data. Adam with weight decay is used to optimize model parameters and establish learning rates in the optimizer, which is loaded from Torch's optim module. The steps include forward propagation, gradient reset, loss computation, backward propagation with parameter updates, and, in the end, using the save function to save model weights and parameters. Scikit-learn handles dataset splitting and model evaluation, re. matches text for regular expressions, OS handles file directories, tqdm shows the training progress bar, and the Pandas and NumPy libraries handle data transformation and storage. The table lists the important experimental parameters.

Table 1

Parameter	Value
roberta_model_name	'roberta-base'
num_classes	2
hidden_size	128
num_filters	100
filter_sizes	[3-5]
lstm_layers	2
dropout	0.5
num_heads	8
RoBERTa Parameter freeze	4
max_len	128
test_size	0.2
batch_size	16
shuffle	True
epochs	3
optimizer	AdamW
learning rate	2e-5
loss function	CrossEntropyLoss

To provide a thorough assessment, we mainly used accuracy, precision, recall, and F1-score in the comparative analysis of experimental outcomes. This method avoids the drawbacks of depending just on one statistic, such as situations where precision is high but recall is low, by measuring the model's capacity to identify a particular category holistically. The following are the formulas for each evaluation metric: parameters are displayed in the table.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$F1 = \frac{2*\text{Precision}*\text{Recall}}{\text{Precision} + \text{Recall}}$$

TP stands for the number of samples that are both positively predicted by the model and positively actually occur; TN stands for the quantity of samples that are both truly negative and that the model also expected to be negative; FP is the number of samples that the model mistakenly predicts as positive but are actually negative; The number of samples that are genuinely positive but that the model mispredicted as negative is denoted by FN. These metrics allow for a thorough assessment of the model's performance in terms of generalization ability, fitting accuracy, category identification detail, and usefulness in real-world applications.

The table displays the experimental findings. The results showed that all RoBERTa-based models outperformed the BERT and Word2Vec series by achieving test accuracies above 94.92%. This benefit results from RoBERTa being an

Table 2

Evaluation Metrics	Overall	Negative Category	Positive Category
Precision	0.96	0.94	0.97
Recall	0.95	0.97	0.94
F1 Score	0.95	0.96	0.95

Table 3

Model Name	Test Accuracy
RoBERTa-CNN-BiLSTM-Attention	0.9549
RoBERTa-CNN-BiLSTM	0.9527
RoBERTa-BiLSTM-Attention	0.9529
RoBERTa-BiLSTM	0.9492
BERT-CNN-BiLSTM-Attention	0.9485
BERT-BiLSTM	0.9377
word2vec-CNN-BiLSTM-Attention	0.9327
word2vec-BiLSTM	0.9174

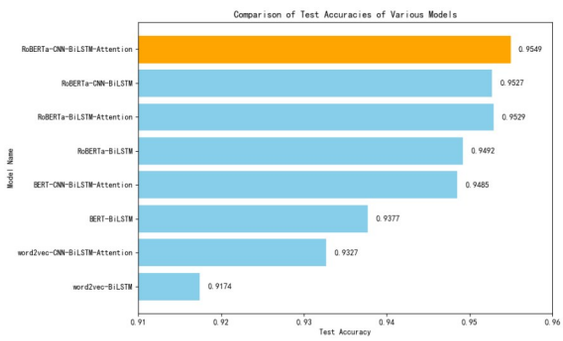

Figure 2

improved version of BERT. RoBERTa exhibits enhanced capabilities as a foundational feature extractor by better capturing deep textual semantics through the optimization of its pre-training technique. Word2Vec lacks contextual dependency information and is unable to handle polysemy because it is a static word vector model. Even when combined with sophisticated structures, this leads to restricted basic feature expression capabilities, making it challenging to outperform pre-trained language models such as RoBERTa and BERT.

The Attention mechanism produces notable improvements under the same fundamental model. When compared to models without the Attention mechanism, those with it perform better. For example, RoBERTa-CNN-BiLSTM-Attention performs better than RoBERTa-CNN-BiLSTM, and RoBERTa-BiLSTM-Attention performs better than RoBERTa-BiLSTM. The strengths of CNN and BiLSTM are complementary; CNN is good at extracting local features while BiLSTM excels at capturing sequence dependencies. The diversity of features is improved by their combination. For instance, the model that uses only BiLSTM performs the worst, whereas the RoBERTa-CNN-BiLSTM-Attention model outperforms the RoBERTA-BiLSTM-Attention model. This illustrates how difficult it is for a single architecture to adequately extract textual information. There is no discernible category bias in the models, as evidenced by all models' precision, recall, and F1 scores being evenly distributed between two groups with comparable support rates.

According to these experiments, the best model for this investigation is RoBERTa-CNN-BiLSTM-Attention. Its main advantages include using RoBERTa as a basis to provide strong context-dependent features; combining CNN and BiLSTM to cover text information at various granularities; and adding an Attention method to increase the weight of important sentiment words. In conclusion, it performs exceptionally well in binary classification tasks with a test accuracy of 95.49%.

4 Conclusion

The RoBERTa-CNN-BiLSTM-Attention model is a hybrid architecture for text sentiment analysis that combines deep learning elements with pre-trained language models. Comparative tests against several baseline models are used to verify its efficacy. It demonstrates the effectiveness of multi-component collaborative modeling by outperforming other models with a test accuracy of 95.49% on the Amazon product review dataset. As the foundational encoder, RoBERTa forms the essential support for model performance by offering richer contextual semantic representations than BERT and Word2Vec through its optimized pre-training technique. The combination of CNN's proficiency in capturing phrase-level local sentiment features and BiLSTM's ability to represent bidirectional sequence dependencies in lengthy texts greatly increases feature diversity. While global attention pooling concentrates on important sentiment terms and efficiently filters out unnecessary information to increase the model's sensitivity to core sentiment signals, the attention technique improves interactive modeling of sequence features.

Even if the suggested model performs exceptionally well in testing, there is still opportunity for improvement and growth. High computational complexity could result from hybrid models' multi-level architecture. In order to speed up inference while preserving performance and making it more appropriate for real-time settings, future research could investigate lightweight designs like model pruning, knowledge distillation, or the introduction of dynamic routing

techniques. This study can be expanded to Chinese and multilingual sentiment analysis based on English product review data. To improve generalization capabilities, models can be tailored for certain domains via transfer learning or domain-adaptive pre-training. Future research can extend to fine-grained sentiment analysis, like five-star ratings, sentiment intensity prediction, or sentiment reason extraction, in conjunction with attribute-level sentiment modeling to enhance practical utility, even though the current model concentrates on binary classification (positive/negative). Techniques like data augmentation and adversarial training help improve the model's resilience to interference and guarantee steady performance in complicated contexts by addressing frequent text noise and adversarial instances in real-world scenarios.

In conclusion, this study provides fresh perspectives on text semantic understanding while validating the efficacy of hybrid structures in sentiment analysis. It is anticipated that multidimensional optimization and expansion would further develop the usefulness of sentiment analysis technologies in actual business situations.

References

[1] Wankhade, M., Rao, A. C. S., & Kulkarni, C. (2022). A survey on sentiment analysis methods, applications, and challenges. Artificial Intelligence Review, 55(7), 5731-5780.

[2] Mejova, Y. (2009). Sentiment analysis: An overview. University of Iowa, Computer Science Department, 5, 1-34.

[3] Liu, B. (2022). Sentiment analysis and opinion mining. Springer Nature.

[4] Darwich M, Mohd S A, Omar N, et al. Corpus-Based Techniques for Sentiment Lexicon Generation: A Review[J]. J. Digit. Inf. Manag., 2019, 17 (5): 296.

[5] Darwich, M., Mohd, S. A., Omar, N., & Osman, N. A. (2019). Corpus-Based Techniques for Sentiment Lexicon Generation: A Review. J. Digit. Inf. Manag., 17(5), 296.

[6] Ahmad, M., Aftab, S., Muhammad, S. S., & Ahmad, S. (2017). Machine learning techniques for sentiment analysis: A review. Int. J. Multidiscip. Sci. Eng, 8(3), 27.

[7] Dey, L., Chakraborty, S., Biswas, A., Bose, B., & Tiwari, S. (2016). Sentiment analysis of review datasets using naive bayes and k-nn classifier. arXiv preprint arXiv:1610.09982.

[8] Ahmad, M., Aftab, S., & Ali, I. (2017). Sentiment analysis of tweets using svm. Int. J. Comput. Appl, 177(5), 25-29.

[9] Xie, X., Ge, S., Hu, F., Xie, M., & Jiang, N. (2019). An improved algorithm for sentiment analysis based on maximum entropy. Soft Computing, 23 (2), 599-611.

[10] Kolchyna, O., Souza, T. T., Treleaven, P., & Aste, T. (2015). Twitter sentiment analysis: Lexicon method, machine learning method and their combination. arXiv preprint arXiv:1507.00955.

[11] Yadav, A., & Vishwakarma, D. K. (2020). Sentiment analysis using deep learning architectures: a review. Artificial Intelligence Review, 53(6), 4335-4385.

[12] Guo, B., Zhang, C., Liu, J., & Ma, X. (2019). Improving text classification with weighted word embeddings via a multi-channel TextCNN model. Neurocomputing, 363, 366-374.

[13] Can, E. F., Ezen-Can, A., & Can, F. (2018). Multilingual sentiment analysis: An RNN-based framework for limited data. arXiv preprint arXiv: 1806.04511.

[14] Xu, G., Meng, Y., Qiu, X., Yu, Z., & Wu, X. (2019). Sentiment analysis of comment texts based on BiLSTM. Ieee Access, 7, 51522-51532.

[15] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.

[16] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019, June). Bert: Pre-training of deep bidirectional transformers for language understanding. 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) (pp. 4171-4186).

[17] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

[18] Ermakova, T., Fabian, B., Golimblevskaia, E., & Henke, M. (2023). A comparison of commercial sentiment analysis services. SN Computer Science, 4(5), 477.

[19] He, A., & Abisado, M. (2024). Text sentiment analysis of Douban film short comments based on BERT-CNN-BiLSTM-Att model. Ieee Access, 12, 45229-45237.

[20] Deng, J., & Liu, Y. (2025). Research on sentiment analysis of online public opinion based on RoBERTa – BiLSTM – attention model. Applied Sciences, 15(4), 2148.

[21] Yin, X., Huang, Y., Zhou, B., Li, A., Lan, L., & Jia, Y. (2019). Deep entity linking via eliminating semantic ambiguity with BERT. IEEE Access, 7, 169434-169445.

[22] Rehman, A. U., Malik, A. K., Raza, B., & Ali, W. (2019). A hybrid CNN-LSTM model for improving accuracy of movie reviews sentiment analysis. Multimedia Tools and Applications, 78(18), 26597-26613.

[23] Rhanoui, M., Mikram, M., Yousfi, S., & Barzali, S. (2019). A CNN-BiLSTM model for document-level sentiment analysis. Machine Learning and Knowledge Extraction, 1(3), 832-847.

[24] Chen, P., Sun, Z., Bing, L., & Yang, W. (2017, September). Recurrent attention network on memory for aspect sentiment analysis. In Proceedings of the 2017 conference on empirical methods in natural language processing (pp. 452-461).