Berger Scientfic Press

Innovation & Technology Advances, 2024, 2(2), 34-45
https://bergersci.com/index.php/jta/index
DOI: 10.61187/ita.v2i2.65

Multiscale Deep Neural Networks: Unveiling New Directions in Text Sentiment Analysis

Hongyu Hu[1,*], Jie Zhang[2], Yang Sun[3]

[1]Mental Health Center, Wuhan Donghu University, No.301 Wenhua Street, 430200, Wuhan, Hubei, China
[2]College of Medicine and Biological Information Engineering, Northeastern University, No.500 Wisdom Street, 110169, Shenyang, Liaoning, China
[3]College of Life Sciences, Shandong Normal University, No.88 Wenhua East Road, 250014, Jinan, Shandong, China

*Correspondence: huhongyu525@163.com

Abstract: The rapid proliferation of textual data across online platforms necessitates accurate sentiment analysis. Traditional sentiment analysis methods, which are based on lexical ontology theories and basic rules, have shown limitations in capturing the subtleties and contextual nuances of language. Recent advancements in machine learning and deep learning have shifted the focus toward model-based approaches, yet they often overlook distinct emotional dimensions in varying text structures. To address this issue, we introduce a novel deep neural network architecture that employs multiscale feature extraction and is designed to capture a broad series of emotional features within texts. This approach significantly improves the accuracy of sentiment analysis by effectively discerning subtle emotional nuances. We validate the effectiveness of our proposed model through extensive experiments and comparisons with benchmark methods, demonstrating its superiority in sentiment analysis tasks. Additionally, a detailed ablation study highlights the impact of the multiscale module on the model's performance.

Keywords: textual data; sentiment analysis; deep learning; multiscale feature; term frequency-inverse document frequency

How to cite this paper: Hu, H., Zhang, J., Sun, Y. The Multiscale Deep Neural Networks: Unveiling New Directions in Text Sentiment Analysis. Innovation & Technology Advances, 2024, 2(2), 34–45. https://doi.org/10.61187/ita.v2i2.65

1. Introduction

In the digital era, the daily generation of textual data, including online community posts, product reviews, movie critiques, and expressions of teenage angst, is increasing at an unprecedented rate. These texts, which are rich in meaningful emotional tone, play an increasingly crucial role across various domains, from social media interactions to psychological assessments [1–8]. Accurately identifying sentiment is meaningful, as it can aid businesses in understanding consumer emotions and refining their strategic operations and play a key role in preventing mental illness in adolescents. Additionally, its influence extends to areas such as public opinion assessment and customer service. Nevertheless, accurate identification of emotions in text continues to be a challenging task.

Traditional text sentiment analysis methods primarily rely on lexical ontology theories and basic rules. For instance, Ali et al. proposed a sentiment classification method combining ontology and topic modeling using latent Dirichlet allocation and word embeddings, effectively filtering out irrelevant content to extract pertinent information [9]. Zhuang et al. investigated the feasibility of knowledge-driven sentiment analysis in terms of efficiency and effectiveness [10]. Colace et al. also adopted the probabilistic approach of latent Dirichlet allocation as an emotion capturer, demonstrating its efficacy [11]. This technique facilitates the automatic extraction of term co-occurrence graphs. Mevskele et al. introduced a bidirectional contextual attention mechanism to evaluate the impact of each word in a sentence on sentiment value [12]. Kontopoulos et al. performed sentiment analysis on Twitter posts [13], innovatively obtaining sentiment scores for distinct concepts within the posts. Additionally, Dragoni et al. proposed a common-sense ontology for sentiment analysis based on SenticNet [14], a semantic network of 100,000 concepts rooted in conceptual primitives. Despite their wide application, these methods often struggle to fully grasp the subtleties and contextual information of language.

The advent of machine learning has shifted the focus of research toward model-based sentiment analysis. Neethu et al. were pioneers in applying machine learning methods to analyze Twitter posts about electronic products such as mobile phones and laptops [15]. Agarwal et al. explored machine learning algorithms combined with feature selection techniques to manage the high-dimensional feature space of the bagof-words model [16]. Hasan et al. utilized supervised machine learning algorithms such as naive Bayes and support vector machines for political view analysis [17]. In addition to machine learning, deep learning has gained prominence in this field, with convolutional neural networks (CNNs) and recurrent neural networks (RNNs) being particularly notable. These models have shown significant potential in extracting text features and deciphering complex semantics. Rhanoui proposed a hybrid of convolutional neural networks and bidirectional long short-term memory (BiLSTM) models for sentiment analysis of extensive texts [18]. Agarwal et al. suggested four different RNN variants to analyze speakers' speech in videos [19].

However, these approaches still have certain limitations. Specifically, they often focus on fixed-scale feature extraction, potentially resulting in an oversight of distinct emotional dimensions across expressions of varying lengths and structures within the text. Therefore, this paper introduces a deep neural network approach based on multiscale feature extraction for emotion detection in textual content. The multiscale network adeptly captures various granular information within the text, thereby enhancing the precision of comprehending and categorizing subtle emotional nuances.

The primary contributions of this paper can be summarized as follows:

1. We propose a deep learning architecture that integrates multiscale feature extraction to effectively capture emotional features within text.
2. We conduct a series of experiments, comparing the proposed model against various benchmark methods and validating its effectiveness and superiority in sentiment analysis tasks.
3. A detailed ablation study was performed to assess the impact of the multiscale module on the model performance.

2. Materials and Methods

In this section, we comprehensively detail the development and application of our model. The introduction is methodically divided into the following subsections: Overview, Dataset, Data Transformation, Multiscale Feature Extraction, Model Training, and Model Evaluation.

The collected text is subjected to preprocessing and tokenization for the purposes of data conversion and feature transformation. These tokens are then divided into two separate processing streams. The top panel shows the multiscale module, while the bottom panel shows the TFIDF module.

2.1. Overview

Figure 1 depicts the comprehensive framework of our study. The collected text is subjected to preprocessing and tokenization for the purposes of data conversion and feature transformation. These tokens are then divided into two separate processing streams. In the multiscale module, tokens undergo a lookup table operation with the embedding layer's weight matrix, converting the index into its corresponding embedding vector, thus forming a textual embedding representation. This representation is subjected to multiple convolutional kernels, which act as filters, to extract features at various scales. The features obtained from these scales are merged and passed to the final deep learning classifier. Meanwhile, in the TF-IDF module, the original tokens are converted into a data matrix, which is subsequently input into four traditional machine learning classifiers.

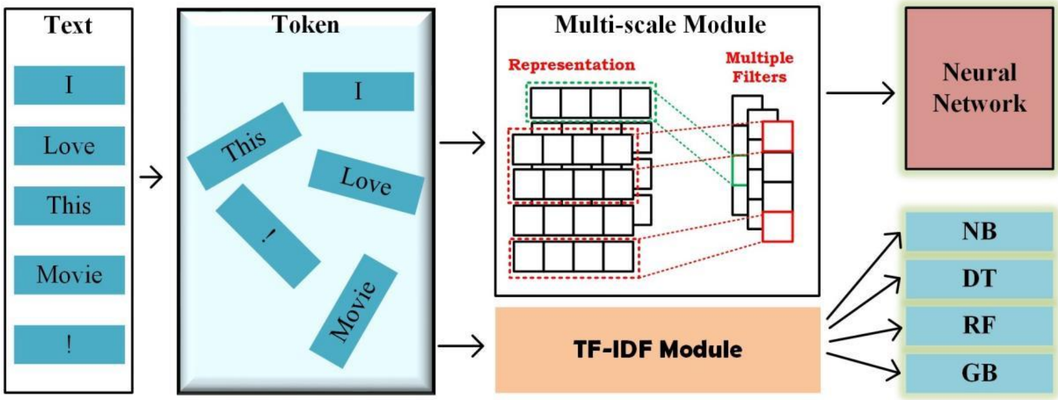


Figure 1. A brief introduction to our framework.

2.2. Dataset

This article utilizes a publicly available dataset [20], initially compiled by Maas et al., for sentiment classification evaluation. This dataset contains movie reviews, each tagged as either positive or negative. To provide a representative sample, we randomly chose 10 entries from the dataset, as displayed in **Table 1**. Due to the considerable length of these reviews, we selected specific excerpts and omitted the remaining content. Additionally, we performed quality control on the original dataset, removing any low-quality samples. The selection process after quality control was randomized to maintain objectivity. As a result, we compiled a text classification dataset comprising approximately 5000 high-quality samples. For conciseness, we will refer to this dataset as the Processed internet Movie (PIM) dataset in subsequent sections of the article.

Table 1. Contents of the dataset: Movie Reviews and Sentiments

Review	Sentiment
I can't believe it! I thought this is a good sequel...	Negative
Impressed! This is the worst SRK movie and one of the worst Bollywood movies I ever saw!...	Negative
More like psychological analysis of movies, but Psycho does sound better as a header...	Positive
This is a straight-to-video movie, so it's not going to rival the first Lion King...	Positive
I saw this movie yesterday and thought it was awful; it was pointless and just plain stupid...	Negative
Do not be mistaken, this is neither a horror, nor truly a film...	Negative
I usually don't categorize a moving as boring...	Negative
Once again Jet Li brings his charismatic presence to the movie screen in the film Black Mask...	Positive
I rented this movie, after hearing Chris Gore saying something to the effect of "five stars!"...	Negative
This was a big disappointment for me...	Negative

2.3. Data Transformation

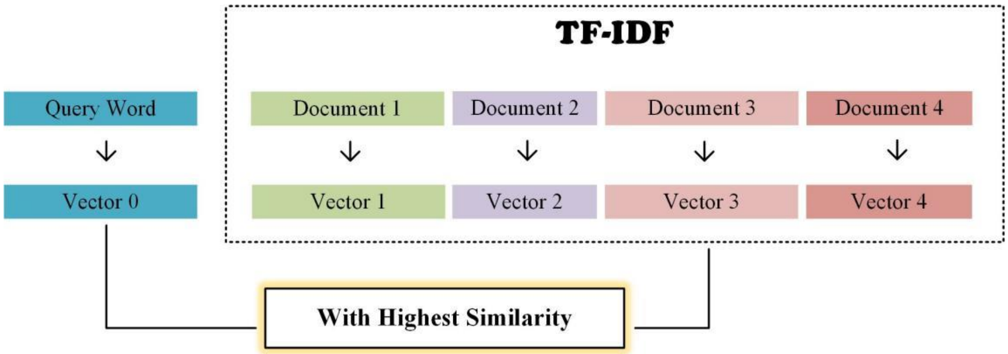


Figure 2. A brief introduction to the term frequency-inverse document frequency (TF-IDF), which is a numerical statistic used in text analysis. It reflects the importance of a word in a document in a collection or corpus.

The initial text file we obtain is represented as $T = "t_1 t_2 \ldots t_m"$, exemplified by the phrase “I love this world!”, where t_i represents the i-th word input. In natural language processing (NLP), it is a standard practice to convert text into a “token” format to improve computer understanding and processing efficiency. This procedure involves breaking down the original text into individual words, phrases, or characters. The tokenization process can be mathematically formulated as follows:

$$\mathrm{tokenize}(\mathrm{T}) = [\mathrm{t_1 t_2 \ldots t_m}] \quad (1)$$

Then, each token t_i is correspondingly mapped to a unique index i_k:

$$\mathrm{map}(t_i) = i_{k_1} \text{ for } i = 1{,}2 \ldots, \mathrm{m} \quad (2)$$

Let **E** represent the weight matrix of nn. The embedding layer is characterized by dimensions $N \times D$. Here, N denotes the size of the vocabulary, and D symbolizes the dimension of the embedding vector. Consequently, the definition of **E** is as follows:

$$\mathbf{E} = \begin{bmatrix} \mathrm{e}_1^{\mathrm{T}} \\ \mathrm{e}_2^{\mathrm{T}} \\ \vdots \\ \mathrm{e}_{\mathrm{N}}^{\mathrm{T}} \end{bmatrix}_{N \times D} \quad (3)$$

convert the index i_k, obtained from Equation (2), into the corresponding embedding vector using the lookup table **E**. The process is described as follows:

$$\mathrm{embed}(i_k) = \mathbf{E}_{i_k}, \text{ for } i = 1{,}2 \ldots, \mathrm{m} \quad (4)$$

By concatenating these resulting embeddings, the embedding representation of the token sequence **Emb** for the original text **T** is obtained:

$$\mathbf{Emb} = [\mathrm{embed}(i_1), \mathrm{embed}(i_2), \ldots, \mathrm{embed}(i_m),] \quad (5)$$

This embedding is subsequently input into a deep neural network trained for emotion recognition in text. Details of this learning process are described in subsequent sections. It is noteworthy that the feature transformation approach in deep learning significantly differs from that in traditional machine learning. In the absence of a neural network in the machine learning model, the term frequency-inverse document frequency (TF-IDF) method was employed for text data transformation into a feature matrix. This statistical method for text feature extraction represents each document as a vector, where each element corresponds to a word, and its value is the word’s TF-IDF score in the document. The methodology is based on the principles of term frequency and inverse document frequency. Term frequency (TF) quantifies the frequency of a word’s appearance in a document, while inverse document frequency (IDF) measures the word’s importance across the entire corpus. The computation process and respective formulae are as follows:

1. **Term frequency (TF):** This measures how frequently a term occurs in a document. The TF for a term t in a document d is calculated as

$$TF(t,d) = \frac{\text{Number of times term } t \text{ appears in document } d}{\text{Total number of terms in documnet } d}$$

2. **Inverse Document Frequency (IDF):** This measures the importance of the term in the entire corpus. The IDF for a term *t* is calculated as:

$$IDF(t) = \log \frac{\text{Total number of document}}{\text{Number of documnets with term } t \text{ in them}}$$

3. **TF-IDF Score:** The TF-IDF score is the product of TF and IDF, representing the relative importance of a term in a document, considering the entire corpus:

$$TF - IDF(t,d) = TF(t,d) \times IDF(t)$$

2.4. Multiscale Feature Extraction

Text data contain information at multiple levels and scales, including word-level, phrase-level and sentence-level semantics. Moreover, different sentiment analysis tasks may require attention to different levels of information. This means that using fixed-scale text feature extractors may not be appropriate. Therefore, in this work, a multiscale-based learning network is proposed to learn text features. Specifically, we achieve this by designing convolution kernels $[f_1, f_2, \ldots, f_p]$ with different scales. For a specific convolution kernel f_k, we first apply a convolution layer and a nonlinear activation function to complete the feature mapping:

$$\mathbf{C_k} = \mathrm{ReLU}(\mathrm{Conv1d_k}(\mathbf{Emb})) \in \mathbb{R}^{m \times n_f} \tag{6}$$

The resulting feature matrix is max-pooled for downstream analysis:

$$\mathbf{P_k} = \mathrm{MaxPool}(\mathbf{C_k}) \in \mathbb{R}^{n_f} \tag{7}$$

The features under the pooled multiple convolution kernels are spliced to achieve multiscale information acquisition.

$$\mathbf{F} = \mathrm{Concatenate}([\mathbf{P_1}, \mathbf{P_2}, \ldots, \mathbf{P}_{\boldsymbol{n_f}}]) \tag{8}$$

The overfitting of spliced features is reduced through dropout regularization technology:

$$\mathbf{F}_{\mathrm{dropout}} = \mathrm{Dropout}\ (\mathbf{F}) \tag{9}$$

Finally, we send the obtained multiscale features, namely, $\mathbf{F}_{\mathrm{dropout}}$, into the downstream neural network for training and analysis.

2.5. Model training

This network is composed of layers, where each layer, denoted as *i*, comprises a linear transformation $\boldsymbol{z}^{(i)}=\boldsymbol{W}^{(i)}\boldsymbol{h}^{(i-1)}+\boldsymbol{b}^{(i)}$ and a nonlinear activation function $g^{(i)}$. Here, $h^{(0)}=x$ represents the input, and $\boldsymbol{h}^{(i)}=g^{(i)}(\boldsymbol{z}^{(i)})$ signifies the output of layer *i*. The weights and biases of each layer are represented by $\boldsymbol{W}^{(i)}$ and $\boldsymbol{b}^{(i)}$, respectively. In our work, the obtained multiscale feature is $\mathbf{F}_{\mathrm{dropout}}$, so the process is formulated as follows:

$$\mathbf{Z}^{(l)} = \mathbf{W}^{(l)}\mathbf{F}_{dropout}^{(l-1)} + \mathbf{b}^{(l)} \tag{10}$$

In the last layer, we apply the softmax function as follows:

$$\mathbf{Y}_{pred} = \mathrm{softmax}(\mathbf{Z}^{(l)}) \tag{11}$$

to transform the output $\mathbf{Z}^{(l)}$ into a probability vector $\mathbf{Y}_{pred}$, where l is the index of the last layer. During training, the loss function is defined as the cross-entropy loss function:

$$\mathcal{L}(\mathbf{Y}_{pred}, \boldsymbol{t}) = -\sum_{i=1}^{\mathbf{N}_c} t_i \log \mathbf{Y}_{pred_i} \tag{12}$$

2.6. Model evaluation

In artificial intelligence, the evaluation of models is very important. In this paper, we introduce 4 evaluation metrics to conduct a comprehensive evaluation of the model. These metrics are built on the correct classification of samples by the model, where we first calculate four important elements. TP (true positive) indicates the number of samples that the model correctly predicts as positive examples. True negative (TN) indicates the number of samples that the model correctly predicts as negative examples. FP (false positive): Indicates the number of samples in which the model incorrectly predicts negative examples as positive examples. FN (false negative): Indicates the number of samples in which the model incorrectly predicts positive examples as negative examples. After obtaining these four metrics, we calculated the accuracy, precision and recall:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{13}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{14}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{15}$$

Then, we calculate the F1 score based on the values of precision and recall:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{16}$$

Precision Recall F1 Score = 2 × Precision + Recall

3. Results

This section presents a comprehensive assessment of the performance of the developed model. Subsequent chapters will delve into diverse aspects, such as comparison with benchmark methods, ablation studies, hyperparameter tuning, model convergence analysis, and model robustness analysis.

3.1. Comparison with Benchmark Methods

Our model was compared with four established machine learning algorithms, namely, naive Bayes (NB), decision tree (DT), random forest (RF), and gradient boosting (GB), which serve as benchmark methods. The results, presented in **Table 2**, highlight the superior-performing outcomes in bold. Notably, our methodology consistently achieved the highest rank across all four evaluative metrics. Compared to the runner-up method, our approach demonstrated improvements exceeding 2.8% in each metric. Additionally, the random forest model exhibited significant efficacy. As an ensemble model composed of multiple decision trees, its expected superior performance over a single decision tree is logical. Each constituent subdecision tree is constructed from unique subsamples and a random selection of features, allowing each tree to capture different aspects of the data. The combination of these trees enables the random forest algorithm to thoroughly assimilate the complex features and interrelationships in the text. This aligns with the guiding principles of our multiscale network, providing indirect evidence of the importance of multiscale features in text sentiment analysis. In summary, our innovatively designed multiscale feature extraction network achieved poor performance, empirically validating the utility of multiscale features.

Table 2. Comparison of the results of the five models.

Model	Accuracy	Precision	Recall	F1 Score
NB	0.819	0.829	0.819	0.816
DT	0.689	0.689	0.689	0.689
RF	0.840	0.841	0.840	0.840
GB	0.797	0.800	0.797	0.797
Ours	**0.864**	**0.864**	**0.864**	**0.864**

The best metric is highlighted in bold.

3.2. Ablation Study

To validate the conclusions drawn in the preceding section, we conducted detailed ablation experiments, quantitatively examining changes in model efficacy to substantiate the effectiveness of our proposed modules. Specifically, we created an alternative version of the multiscale feature extraction network by replacing the original multiscale module with a fixed-scale counterpart and then evaluated its performance. This modification involved shifting from multiple convolution kernels, used in the initial experiment, to a single convolution kernel for feature extraction at a uniform scale. As shown in **Figure 3**, the integration of the multiscale module significantly enhanced the classification capabilities. Conversely, replacing the multiscale module with a fixed-scale variant resulted in a decrease in performance. The removal of the multiscale module caused a noticeable reduction in efficiency of 7.46%. Combining the findings from **Figure 3** with insights from the previous section, we can confidently assert that textual information may inherently exhibit multiscale characteristics, and our expertly designed multiscale learning network effectively captures this nuanced information.

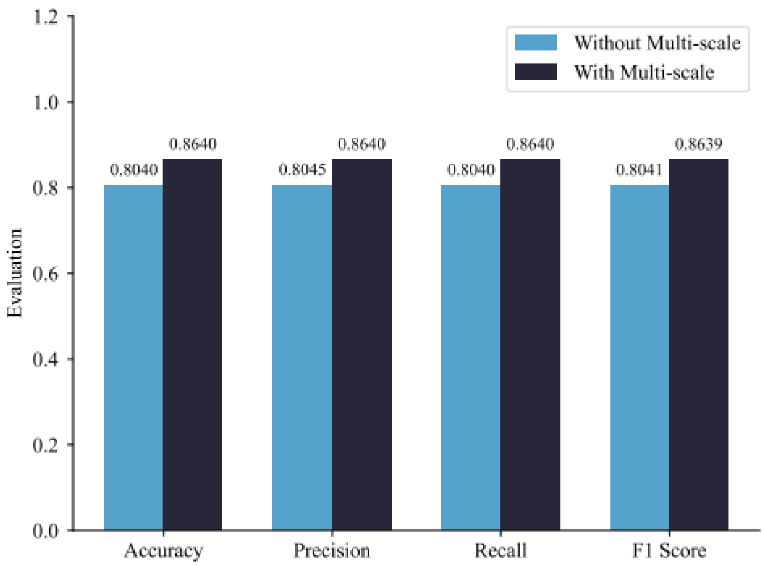
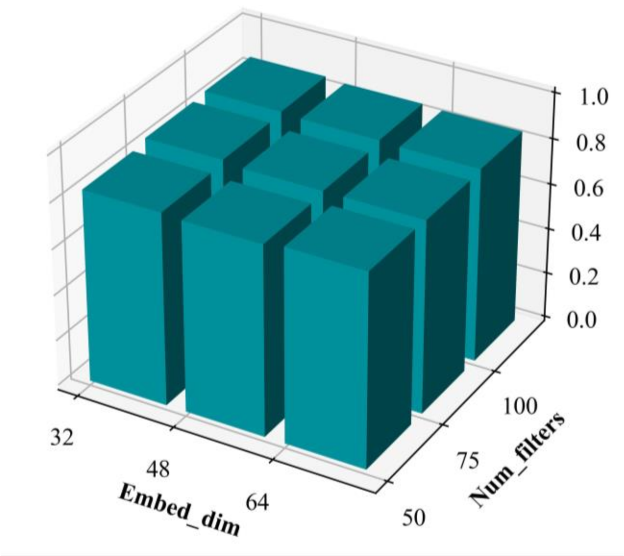


Figure 3. Ablation study for models with and without multiple scales.

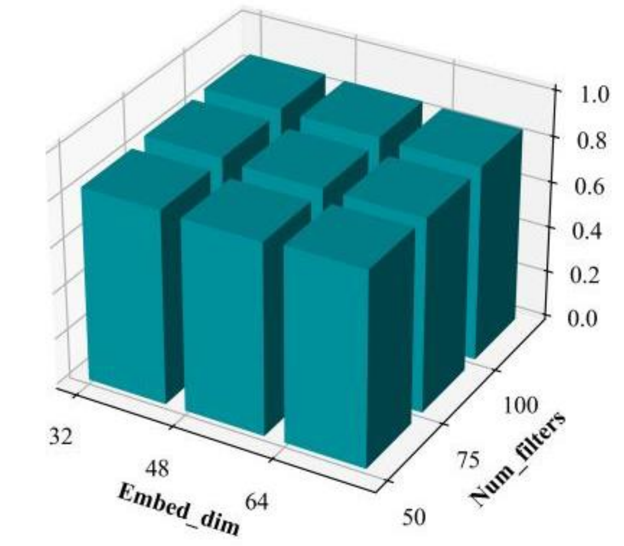
3.3. Hyperparameter Tuning

Throughout the design phase of our model, we kept specific hyperparameters constant, namely, the word embedding dimension and the number of convolutional filters.

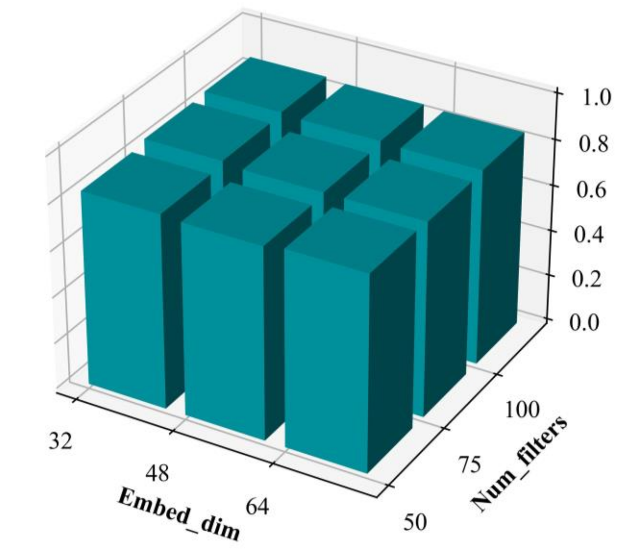
In this chapter, we conducted a parameter search to assess the sensitivity of these hyperparameters. We explored the parameter landscape, investigating embedding dimensions set at [32, 48, 64] and filter counts at [50, 100, 150]. This process resulted in nine unique sets of experimental results for each evaluation metric. To present these results more intuitively, we utilized three-dimensional bar charts, as illustrated in **Figure 4**.



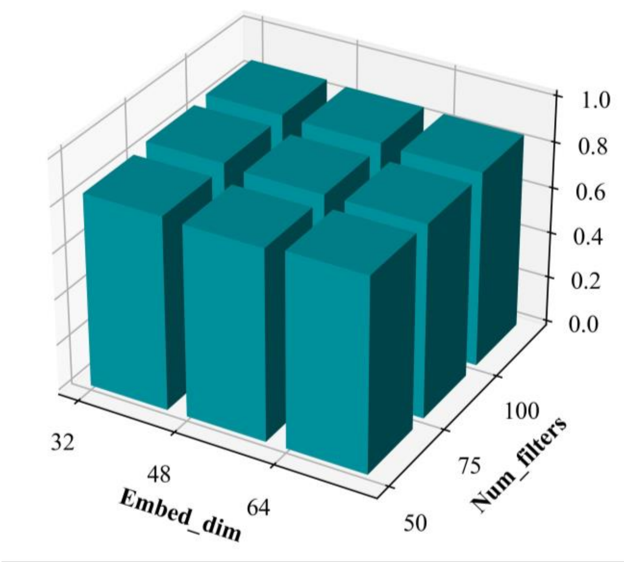
(a)



(b)



(c)



(d)

Figure 4. Experiments assessing the sensitivity of two hyperparameters, Embed dim and Num filters, are depicted. Subfigures (a-d) depict the results evaluated using the accuracy, precision, recall, and F1 score metric.

The graph distinctly shows that both hyperparameters exhibit minimal sensitivity across the four evaluation metrics, suggesting that changes in these parameters do not result in notable performance variations in the model. Simultaneously, a slight advantage was noted when the embedding dimension was set to 32 and the filter count was set to 100 relative to other parameter combinations. Therefore, in this study, we set the word embedding dimension and the number of convolutional filters at 32 and 100, respectively, to optimize model performance. However, users retain the flexibility to adjust these parameters, as our results indicate a relative insensitivity to these values. Furthermore, it was observed that an increase in the number of convolutional filters does not consistently lead to a significant improvement in performance. An excess of filters may lead the model to assimilate noise or excessively localized features from the training data, resulting in feature redundancy instead of the integration of beneficial information. This also increases the time complexity. Therefore, in practical applications, it is essential to carefully select the number of convolutional filters in the development of a multiscale feature extraction network.

3.4. Model Convergence Analysis

To ensure the effective training of our model, this section investigates model convergence. Convergence is a critical aspect of model training and is essential for ensuring robust generalization performance. Analyzing convergence is instrumental in facilitating adjustments to the model's architecture and network parameters, and it is crucial for enabling the timely termination of experiments, thus conserving significant computational resources. To conduct this analysis, we graphically depicted the variation in the cross-entropy loss function with respect to the increasing number of epochs. The findings are presented in **Figure 5**. The graph shows that even with training extended to 1000 epochs, the loss function value plateaued and tended to stabilize at approximately 200 epochs. This observation indicates that the trained model achieved effective convergence. Consequently, in this study, we set the number of epochs to 200. This decision allows for the comprehensive training of multiple models while optimizing computational resources and reducing the risk of overfitting.

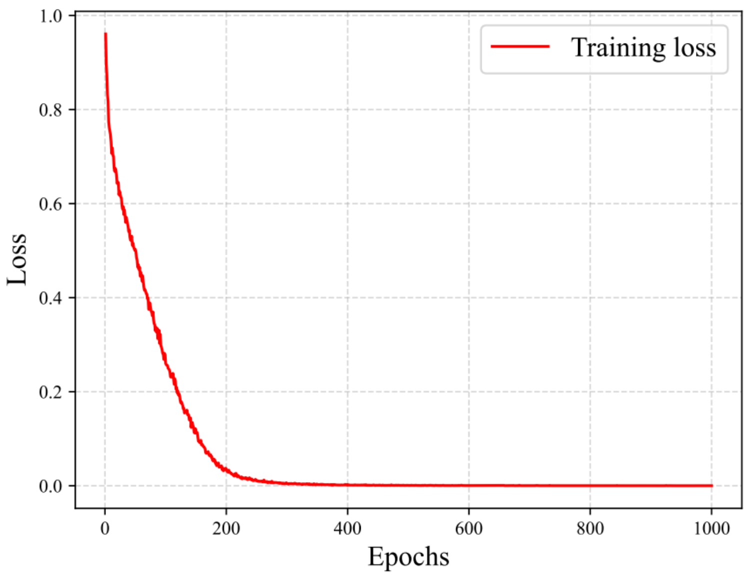


Figure 5. The convergence analysis of the proposed model

3.5. Model Robustness Analysis

The model developed in this study, which is based on deep learning, is subject to various sources of randomness, such as stochastic initialization of initial weights and minor variations in feature transformations. To evaluate the model's robustness, we ran our code five times to monitor performance fluctuations. If the results of multiple iterations show negligible disparities, this indicates that the model is relatively resilient to randomness.

Conversely, significant differences would imply a sensitivity to initial conditions or data inconsistencies. The results from these repeated experiments are compiled in **Table 3**, which displays five sets of metrics across four evaluative parameters. The data reveal that despite minor variances, the model's overall performance is consistently high. To conduct a more rigorous quantitative analysis, we computed the standard deviation of the five consecutive experiments. The results show a low standard deviation, ranging from 0.0108 to a maximum of 0.0110. This demonstrates the model's exceptional robustness, as it maintains a consistent performance standard across repeated trials.

Table 3. Model stability experiments. The model was executed five times with identical parameters to observe data fluctuations. In this table, the last two rows denote the average values and standard deviations of the repeated execution results, respectively.

Model	Accuracy	Precision	Recall	F1 Score
Ours	0.864	0.864	0.864	0.864
Repeat 1 times	0.834	0.835	0.834	0.834
Repeat 2 times	0.855	0.855	0.855	0.855
Repeat 3 times	0.861	0.861	0.861	0.861
Repeat 4 times	0.851	0.851	0.851	0.851
Average	0.853	0.853	0.853	0.853
SD Value	0.0108	0.0110	0.0108	0.0108

4. Discussion

Emotions constitute an integral aspect of human existence, profoundly influencing decision-making and facilitating nuanced communication. Text sentiment analysis, also known as text emotion recognition, employs natural language processing techniques to identify sentiments within text, providing a scientific and comprehensive basis for decision-making relevant to governments, businesses, and consumers. As a result, this field has garnered considerable attention. The primary aim of text-based sentiment analysis is to assess emotions in specific sentences or paragraphs. Traditional approaches use manually crafted features combined with classifiers to produce satisfactory results. Conversely, deep learning models leverage foundational encoders such as RNNs, recursive neural networks (ReNNs), and CNNs, which are enhanced by mechanisms such as attention and memory, thus enhancing the model's capacity to capture complex patterns. To improve the effectiveness of these deep learning models, resources such as syntactic dependency trees, sentiment dictionaries, negation dictionaries, and adverbial degree dictionaries are employed [21–26]. Despite significant progress, the reliance of these methods on fixed-scale feature extractors for feature assimilation often limits the thorough learning of text attributes, thereby affecting classification performance. Furthermore, the poor interpretability and generalizability of neural networks present challenges, as they transform features into a latent space, thus obscuring the mapping process. Additionally, the significant variability of sentiment resources across various domains complicates the cross-domain transfer of textual information.

In this study, we transformed sequential text into representations and utilized multiscale convolutional kernels to rigorously learn multiscale text information. This approach enables the model to capture data at multiple levels and scales, thereby enhancing its proficiency in discerning complex patterns and subtle semantic nuances. The experimental results corroborate the effectiveness of our module. We conducted a comprehensive assessment of the model, examining aspects such as convergence and stability, among others. Additionally, a sensitivity analysis of the parameters was performed to determine the model's parameter stability. This multiscale feature extraction technique allows the model to learn varied and complex representations, thereby enhancing its overall performance in sentiment classification tasks. We aim to adapt the multiscale network to multimodal scenarios and further explore the roles of data, multiscale information extraction, multimodal information fusion, and the derivation of inductive knowledge. Additionally, refining the architecture continues to be a promising direction for future research.

5. Declarations

5.1. Ethical Approval

This study did not involve human participants, human data or tissues, or animals; therefore, no ethical approval was needed.

5.2. Funding

No funding was received for conducting this study. The authors have no financial or proprietary interests in any material discussed in this article.

5.3. Availability of data and materials

The datasets and code can be publicly accessed in the Repository https://github.com/Hongyuu-Hu/MSDNN.

Conflicts of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Zad, S., Heidari, M., Jones, J.H., et al. A survey on concept-level sentiment analysis techniques of textual data. In proceedings of 2021 IEEE World AI IoT Congress (AIIoT), 2021, 0285–0291. https://doi.org/1010.1109/AIIoT52608.2021.9454169
2. Ullah, M.A., Marium, S.M., Begum, S.A.,et al. An algorithm and method for sentiment analysis using the text and emoticon. ICT Express, 2020, 6(4), 357–360. https://doi.org/10.1016/j.icte.2020.07.003
3. Saura, J.R., Palos-Sanchez, P., Grilo, A. Detecting indicators for startup business success: Sentiment analysis using text data mining. Sustainability, 2019, 11(3), 917. https://doi.org/10.3390/su11030917
4. Mouthami, K., Devi, K.N., Bhaskaran, V.M. Sentiment analysis and classification based on textual reviews. In proceedings of 2013 International Conference on Information Communication and Embedded Systems (ICICES), 2013, 271–276. https://doi.org/1010.1109/ICICES.2013.6508366
5. Kauffmann, E., Peral, J., Gil, D., et al. Managing marketing decision-making with sentiment analysis: An evaluation of the main product features using text data mining. Sustainability, 2019, 11(15), 4235. https://doi.org/10.3390/su11154235
6. Pradha, S., Halgamuge, M.N., Vinh, N.T.Q. Effective text data preprocessing technique for sentiment analysis in social media data. In proceedings of 2019 11th International Conference on Knowledge and Systems Engineering (KSE), 2019, 1–8. https://doi.org/1010.1109/KSE.2019.8919368
7. Hu, R., Rui, L., Zeng, P., et al. Text sentiment analysis: A review. In proceedings of 2018 IEEE 4th International Conference on Computer and Communications (ICCC),2018, 2283–2288. https://doi.org/1010.1109/CompComm.2018.8780909
8. Mehta, P., Pandya, S. A review on sentiment analysis methodologies, practices and applications. International Journal of Scientific and Technology Research, 2020, 9(2), 601–609.
9. Ali, F., Kwak, D., Khan, P., et al. Transportation sentiment analysis using word embedding and ontologybased topic modeling. Knowledge-Based Systems, 2019, 174, 27–42. https://doi.org/10.1016/j.knosys.2019.02.033
10. Zhuang, L., Schouten, K., Frasincar, F. Soba: Semi-automated ontology builder for aspect-based sentiment analysis. Journal of Web Semantics, 2020, 60, 100544. https://doi.org/10.1016/j.websem.2019.100544
11. Colace, F., De Santo, M., Greco, L., et al. Probabilistic approaches for sentiment analysis: Latent dirichlet allocation for ontology building and sentiment extraction. Sentiment Analysis and Ontology Engineering: An Environment of Computational Intelligence, 2016, 75–91. https://doi.org/10.1007/978-3-319-30319-2_4
12. Meškelė, D., Frasincar, F. Aldonar. A hybrid solution for sentence-level aspectbased sentiment analysis using a lexicalized domain ontology and a regularized neural attention model. Information Processing & Management, 2020, 57(3), 102211. https://doi.org/10.1016/j.ipm.2020.102211
13. Kontopoulos, E., Berberidis, C., Dergiades, T., et al. Ontology-based sentiment analysis of twitter posts. Expert systems with applications, 2013, 40(10), 4065–4074. https://doi.org/10.1016/j.eswa.2013.01.001
14. Dragoni, M., Poria, S., Cambria, E. Ontosenticnet: A commonsense ontology for sentiment analysis. IEEE Intelligent Systems 2018, 33(3), 77–85. https://doi.org/1010.1109/MIS.2018.033001419

15. Neethu, M., Rajasree, R. Sentiment analysis in twitter using machine learning techniques. In proceedings of 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), 2013, 1–5. https://doi.org/1010.1109/ICCCNT.2013.6726818
16. Agarwal, B., Mittal, N., Agarwal, B., et al. Machine learning approach for sentiment analysis. Prominent feature extraction for sentiment analysis, 2016, 21–45. https://doi.org/10.1007/978-3-319-25343-5_3
17. Hasan, A., Moin, S., Karim, A., et al. Machine learning-based sentiment analysis for twitter accounts. Mathematical and computational applications, 2018, 23(1), 11. https://doi.org/10.3390/mca23010011
18. Rhanoui, M., Mikram, M., Yousfi, S., Barzali, S.: A cnn-bilstm model for document-level sentiment analysis. Machine Learning and Knowledge Extraction, 2019, 1(3), 832–847. https://doi.org/10.3390/make1030048
19. Agarwal, A., Yadav, A., Vishwakarma, D.K. Multimodal sentiment analysis via rnn variants. In proceedings of 2019 IEEE International Conference on Big Data, Cloud Computing, Data Science & Engineering (BCD), 2019, 19–23. https://doi.org10.1109/BCD.2019.8885108
20. Maas, A., Daly, R.E., Pham, P.T., et al. Learning word vectors for sentiment analysis. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, 2011, 142–150.
21. Poria, S., Cambria, E., Winterstein, G., et al. Sentic patterns: Dependency-based rules for concept-level sentiment analysis. Knowledge-Based Systems, 2014, 69, 45–63. https://doi.org/10.1016/j.knosys.2014.05.005
22. Di Caro, L., Grella, M. Sentiment analysis via dependency parsing. Computer Standards & Interfaces, 2013, 35(5), 442–453. https://doi.org/10.1016/j.csi.2012.10.005
23. Gajendrasinh, R.D., Bohara, M.H. Sentiment analysis for feature extraction using dependency tree and named entities. In proceedings of International Conference on Innovations in Information Embedded and Communication Systems (ICIIECS), 2017, 587–592.
24. Nakagawa, T., Inui, K., Kurohashi, S. Dependency tree-based sentiment classification using crfs with hidden variables. In proceedings of Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, 2010, 786–794.
25. Hardeniya, T., Borikar, D.A. Dictionary based approach to sentiment analysis-a review. International Journal of Advanced Engineering, Management and Science, 2016, 2(5), 239438.
26. Pandey, S., Sagnika, S., Mishra, B.S.P. A technique to handle negation in sentiment analysis on movie reviews. In proceedings of 2018 International Conference on Communication and Signal Processing (ICCSP), 2018, 737–743. https://doi.org/10.1109/ICCSP.2018.8524421