Postprint: A Multi-scale Convolutional Neural Network Model Based on Character-Word Vectors for Sentiment Classification of Weibo Comments and Experimental Study

Authors: Zhang Liu, Wang Xiwei, Huang Bo, Liu Yutong

Date: 2023-07-26T00:00:00+00:00

Abstract

[Purpose/Significance] Sentiment classification models for Weibo comments can provide guidance for relevant public opinion monitoring and regulatory authorities in properly managing the development of topic events and public opinion. [Methods/Process] Utilizing a multi-scale convolutional neural network based on character-word vectors, and employing multi-scale convolutional kernels to mitigate the constraints of limited contextual information in Weibo comments, we construct a sentiment classification model for Weibo comments; by crawling data from "Weibo Hot Search Rectification", we verify the feasibility and superiority of the proposed model. [Results/Conclusion] Experimental results demonstrate that the multi-scale convolutional neural network based on character-word vectors performs well on short text classification tasks with limited contextual information, such as Weibo public opinion analysis. This paper provides a more accurate theoretical model and classification method for sentiment classification of Weibo public opinion at the theoretical level, and can better guide public opinion monitoring and regulatory authorities in directing and supervising the sentiment orientation of public opinion at the practical level.

Full Text

Preamble

ChinaXiv Cooperative Journal
Volume 63, Issue 18, September 2019

A Sentiment Classification Model and Experimental Study of Microblog Comments Based on Multi-Scale Convolutional Neural Networks with Character-Word Vectors

Zhang Liu[1], Wang Xiwei[1,2], Huang Bo[3], Liu Yutong[1]
[1]School of Management, Jilin University, Changchun 130022
[2]Research Center for Big Data Management, Jilin University, Changchun 130022
[3]School of Computer Science and Technology, Jilin University, Changchun 130022

Abstract: [Purpose/Significance]

The sentiment classification model for microblog comments can provide guidance for relevant public opinion supervision departments to correctly manage the development of topic events and public sentiment. [Method/Process] Based on multi-scale convolutional neural networks with character-word vectors, this study employs multi-scale convolution kernels to improve the constraints of limited contextual information in microblog comments and constructs a sentiment classification model for microblog comments using multi-scale convolutional neural networks with character-word vectors. By crawling data on the "Weibo Hot Search Rectification" topic, the feasibility and superiority of the model are verified. [Result/Conclusion] Verification results demonstrate that the multi-scale convolutional neural network based on character-word vectors performs well in short text classification tasks with limited contextual information such as Weibo public opinion. This paper provides a more accurate theoretical model and classification method for sentiment classification of Weibo public opinion at the theoretical level, and can better guide public opinion supervision departments to guide and monitor the emotional tendencies of public opinion in practice.

Classification Number: G203
Keywords: Convolutional Neural Network, Character-Word Vector, Multi-Scale Convolution Kernel, Weibo Comments, Sentiment Classification

With the rapid development of social media, an increasing number of internet users express their emotions, opinions, and views through social media platforms. According to Weibo's financial report for the second quarter of 2018, as of June 2018, Weibo's monthly active users totaled 431 million, an increase of 21 million compared to the previous quarter, with mobile devices accounting for 93% and daily active users increasing to 190 million [1]. As an emerging information publishing and social media platform, Weibo possesses characteristics of openness, immediacy, interactivity, and freedom. The convenience provided by mobile terminal apps allows users to express their own views and emotions at any time, making sentiment classification of Weibo comments a new topic of academic interest that can better assist public opinion management departments in guiding and managing public sentiment.

In recent years, scholars at home and abroad have successively conducted research on sentiment classification of public opinion. Foreign scholar Rui NS [2] constructed an aggregated sentiment semantic recognition and opinion propagation influence model, focusing on analyzing the influencing factors of sentiment propagation; Soleymani M [3] defined the multimodal sentiment analysis problem, arguing that multimodal sentiment analysis has significant development potential; Öztürk N [4] used Twitter to collect 23,881 relevant tweets on public opinion about the Syrian refugee crisis and conducted sentiment analysis, with data results showing that the sentiment of Turkish tweets differs significantly from that of English tweets. Domestic scholar Tang Xiaobo [5] constructed a feature ontology to classify feature words and calculated the sentiment polarity and intensity of Weibo product reviews, verifying the method's effectiveness by collecting Weibo data; Ma Songyue [6] used ROST EA tools for sentiment analysis on Douban movie user comments and built a model for testing through regression methods; Liang Xiaomin [7] constructed a public opinion object analysis model and studied the relationship network of public opinion objects, with experimental results showing that the model can effectively identify main public opinion objects and their sentiment words, and can intuitively display netizens' emotional expressions and relationship recognition of public opinion objects over time.

Currently, sentiment classification tasks in natural language processing are generally treated as temporal signals, with performance primarily dependent on text feature representation methods. Tang Huifeng [8] proposed using BiGrams feature representation methods, combined with information gain feature selection methods and support vector machine classifiers, which perform well when the training set is sufficient and the number of features is appropriate. Zhang Dongwen [9] combined Word2Vec and SVMperf tools, using an expanded sentiment dictionary method to achieve good results. However, text feature representation methods mostly rely on temporal information from context. Due to the very limited contextual information in Weibo comments and the presence of a large number of out-of-vocabulary (OOV) words, word-level vectors in Weibo comments are greatly affected by Chinese word segmentation and cannot provide sufficient prior knowledge for accurate sentiment classification, leading to inaccurate sentiment classification of online Weibo public opinion [10].

In recent years, scholars have continuously proposed using convolutional neural networks from the image domain to solve natural language processing tasks. Feng Xingjie [11] proposed combining convolutional neural networks with attention models to avoid dependence on manually constructed sentiment dictionaries, thereby improving the overall automation performance of the model. Yann LeCun, known as the "father of convolutional neural networks" [12], proposed using 70 character-level vectors (26 English letters, 0-9 Arabic numerals, and common symbols) combined with convolutional neural networks for text classification. By relying on the feature extraction capability of convolutional layers, these character-level vectors can extract meaningful features, attracting widespread attention and enabling convolutional neural networks to shine not only in the image domain but also providing new ideas for natural language processing tasks.

Inspired by this, this paper selects two granularities of vectors—character-level and word-level—based on the convolutional neural network model. These two vectors are concatenated into a three-dimensional vector matrix to construct a multi-scale convolutional neural network model based on character-word vectors. This study attempts to address three research questions: constructing a microblog comment sentiment classification model based on multi-scale convolutional neural networks with character-word vectors; verifying and analyzing the classification accuracy of the constructed model through typical topics; and verifying the superiority of the constructed microblog comment sentiment classification model.

1 Related Theory

1.1 Public Opinion Sentiment Classification

Sentiment is the attitude and psychological feeling that humans hold toward objective events or objects, reflecting the relationship between a subject with certain needs, desires, or viewpoints and an object [13]. Sentiment classification of microblog comments refers to mining and analyzing microblog comments to identify their sentiment tendencies and classify sentiment texts as positive or negative evaluations, or to classify and score them according to sentiment intensity [14]. Due to the temporal nature of sentiment classification texts, by introducing an attention mechanism [15], different parts of the text can be assigned different weights for the classification task, thereby training a weight matrix. By combining the LSTM model in recurrent neural networks with attention mechanisms [16], user emotions on hot issues can be selected using microblog comments as the data source.

Currently, mainstream methods for public opinion sentiment classification [17] are divided into four categories:

(1) **Dictionary-based sentiment classification methods**: The core of these methods is "dictionary + rules," using sentiment dictionaries as the main basis for judging sentiment polarity while considering syntactic structures in comments. The performance of such methods depends on the quality of sentiment dictionaries and corresponding judgment rules, both of which require manual intervention, making these methods dependent on prior knowledge and resulting in poor generalization ability.

(2) **Machine learning-based sentiment classification methods**: The core work of these methods is feature engineering, commonly using N-gram features, syntactic features, and TF-IDF features. These methods employ classic classification models such as Naive Bayes and Support Vector Machines. Similarly, these methods still rely on manual feature design and have limited generalization ability.

(3) **Hybrid dictionary and machine learning classification methods**: These methods treat "dictionary + rules" as a special feature, combine it with existing features (N-gram, syntactic features, etc.), select the fi-

nal feature combination, and then use a classifier for classification. This method combines the advantages of both approaches and is currently the most commonly used method for small-scale samples.

(4) **Deep learning-based methods**: These methods first train semantic word vectors from massive comment corpora, then use different semantic composition methods to construct overall comment feature representations using the trained word vectors. These methods perform well with sufficient data but cannot learn enough contextual information when data volume is limited.

1.2 Chinese Word Segmentation and Word Vector Training

Compared with foreign social media platforms such as Twitter, the main difference in Weibo comments is that users employ Chinese, which leads to significant differences from English text research in the sentiment classification field. First, Chinese text has word segmentation issues—spaces in English text are natural and undisputed word delimiters, whereas Chinese word segmentation is a relatively complex problem that often depends on semantic and contextual information. Second, Weibo comments contain a large number of popular terms, loanwords, homophones, misused characters, and internet slang. These out-of-vocabulary (OOV) words make it difficult for traditional sentiment word segmentation algorithms to effectively segment words, and the quality of segmentation directly affects the quality of sentiment word vectors, thereby influencing the performance of sentiment classification models.

With the deepening research on deep neural networks, particularly convolutional neural networks, scholars have found that even when sentences are split into fine-grained units, effective semantics can be learned through the feature extraction function of convolutional neural networks. In English text, research has proven that even completely abandoning English words and segmenting English into character granularity can achieve good text analysis results [12]. This method splits English into theoretically minimal units, relying on continuous feature extraction and combination by convolutional layers to ultimately form effective semantics. Based on these characteristics, this paper introduces character-level vector representation to maximally preserve the original semantics of Weibo comments, decomposing OOV words into characters. By relying on convolutional layers for effective feature extraction of both character-level and word-level vectors, interference from information loss due to word segmentation on downstream models is prevented.

With the widespread application of deep learning in sentiment analysis of public opinion, an increasing number of researchers have begun constructing shallow neural network models based on probability. The word vectors trained by these models are highly correlated with language models. Through layer-by-layer calculation of neural networks, the model ultimately outputs a low-dimensional vector, with the semantics contained in vocabulary dispersed and stored in various dimensions of the vector. This construction method reduces the dimensionality of vocabulary itself, avoiding the dimensionality disaster problem of one-hot encoding and other models. Among these shallow neural network models, the distributed word vector model Word2Vec proposed by Google [18] is currently the mainstream word vector representation algorithm. By continuously adjusting word vectors based on the contextual semantics of vocabulary constructed in the training corpus, semantically similar words with similar contexts in the corpus have similar vector representations [19]. This paper selects the Word2Vec model for word vector modeling.

1.3 Convolutional Neural Networks

Convolutional Neural Networks (CNN) are typically used in image processing for image feature extraction. With deepening research, convolutional neural networks have gradually been applied to text feature selection [20]. Text data corresponds to massive weights, and through the operations of convolutional and pooling layers in CNNs, effective feature selection can be performed [21]. From the perspective of sentiment tendency analysis, this can eliminate redundant information in text and extract key sentiments.

Convolutional neural networks mainly include two processes: convolution and pooling [22]. Convolution serves the purpose of feature selection, where self-learned "convolution kernels" scan the entire document in the form of a sliding window (for convenience of representation, words in sentiment texts are converted to numbers). Important semantic information is amplified during this process, while non-important semantic information is reduced. The pooling process is a feature dimensionality reduction process aimed at reducing computational load. In sentiment tendency analysis, maximum pooling is typically selected to retain the most critical semantic information. With convolution and pooling, convolutional neural networks can extract important features from sentiment texts while significantly reducing dimensions, thereby increasing computational convenience.

2 Multi-Scale Convolutional Neural Network Model for Weibo Comment Sentiment Classification Based on Character-Word Vectors

2.1 Model Construction Process

Based on the above theoretical foundations, this paper proposes a multi-scale convolutional neural network sentiment classification model based on character-word vectors (Multi-Scale Convolutional Neural Network, Multi-CNN), as shown in [Figure 1: see original paper]. The model construction sequence includes six steps: Weibo comment data collection and preprocessing, character-word level vector training and selection, convolutional neural network training, performance evaluation and model selection, Weibo comment sentiment classification, and data classification result analysis.

1. **Weibo comment data collection and preprocessing**: Primarily obtaining Weibo comment data through web crawlers, using the Jieba segmentation tool to segment words with characters and words as the minimum segmentation units, and performing stop-word removal operations for preprocessing.
2. **Character-word level vector training and selection**: Using Word2Vec to train word vectors, selecting optimal word vectors through cosine similarity, and finally combining character vectors to construct a three-dimensional text matrix.
3. **Convolutional neural network training**: Primarily through feature extraction by convolutional layers and dimensionality reduction by pooling layers, mapping the three-dimensional text matrix to a one-dimensional vector for the fully connected layer to operate on, then training the model and adjusting model parameter weights through backpropagation algorithms on the training set, and performing sentiment word selection through cross-validation.
4. **Performance evaluation and model selection**: Comparing and verifying by calculating accuracy, recall, and F1 values, and selecting the model with the best comprehensive performance as the final classification model.
5. **Weibo comment sentiment classification**: The model performs sentiment classification of Weibo comments based on the finally determined optimal model.
6. **Classification result analysis**: Finally discussing and analyzing the determined Weibo sentiment classification results.

2.2 Text Vector Training and Selection

After data collection and preprocessing, the main problem that Weibo sentiment classification needs to solve is the vectorization representation of sentiment texts, transforming them into computable data for use by sentiment classification models. The general solution is to use word embedding models to map each word to a low-dimensional vector, thereby representing sentiment texts as a matrix composed of these vectors [23]. This study uses Google's open-source Word2Vec word vector training tool to map each character or word after segmentation processing into a vector, using cosine similarity as an evaluation metric for word-level vectors, as shown in formula (1):

$$sim(x,y) = cos\theta = \frac{\vec{x}\cdot\vec{y}}{\|x\|\|y\|} \quad \text{formula (1)}$$

The approach to judging the quality of word-level vectors is that in high-quality word vectors, similar words map to vectors with high cosine values; conversely, unrelated words have smaller cosine values. This paper selects two groups of word-level vectors with optimal quality based on cosine similarity, then combines them with one group of character-level vectors to construct a three-dimensional text matrix, as shown in [Figure 2: see original paper].

2.3 Overall Architecture and Algorithm Flow

After sentiment texts are constructed into a three-dimensional text matrix, text feature extraction and dimensionality reduction are performed through convolutional neural networks. As shown in [Figure 3: see original paper], the initial dimension of the text matrix is 200. After processing by the convolutional layer, although text features can be effectively extracted, the matrix dimension remains 198 without significant reduction. After the pooling layer, the matrix dimension rapidly decreases to 64, which aligns with the functions of different operations in convolutional neural networks: the convolution operation aims to extract text features, while the pooling operation aims to control dimensions. The alternating use of convolutional and pooling layers enables the model to effectively extract text features while reducing dimensions.

As shown in [Figure 4: see original paper], this paper designs three different scales of convolution kernels based on AlexNet grouped convolutions, namely 3$\times 128, 4\times 128, and 5\times$128, to facilitate text feature extraction of three, four, or five adjacent words in sentiment texts. Each type of convolution kernel is set to 128, combined with the same max pooling operation to form a convolutional feature extraction layer. Finally, the three convolution units are combined into a one-dimensional vector.

After feature extraction, the text matrix passes through two fully connected layers. The first fully connected layer has 128 neurons, and the second has 64 neurons, which can stretch the feature-extracted text matrix into a two-dimensional vector again. Because the use of fully connected layers greatly increases the number of model parameters and may even cause overfitting, a Dropout layer must be added to control the number of randomly deactivated neurons in the fully connected layer. Dropout randomly "kills" a certain proportion of neurons in each iteration, specifying their output as zero, so that their connected weights do not participate in weight updates during the backpropagation training process. This controls the number of model parameters, facilitates computation, and also breaks unnecessary dependencies between certain weights, reducing the risk of model overfitting and significantly improving accuracy on the test set.

The essence of the convolutional neural network training phase is an iterative process of calculating the loss function and continuously updating weights according to the backpropagation algorithm [24]. This paper selects the binary cross-entropy loss function as the cost function, with the loss function expression shown in formula (2):

$$\sum_{x} y \ln a + (1-y)\ln(1-a) \quad \text{formula (2)}$$

where y is the expected output, a is the actual output of the neuron, and the overall loss is J. The cross-entropy loss function has good functional properties: When the expectation and actual output are close (i.e., both are 1 or both are 0), the loss J approaches 0. When the expectation and actual output are far apart (i.e., expectation is 0 and actual output is 1, or expectation is 1 and actual output is 0), the loss approaches infinity. Since neural network parameter updates depend on gradient descent, larger losses mean model parameter updates, while zero loss indicates model convergence, meaning expectation and actual output are consistent.

Gradient calculation is shown in formula (3):

$$\sum_x x(\sigma(z) - y)$$

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad \text{formula (4)}$$

where w represents weights and z represents the input of each layer of neurons. Weight updates do not have the σ (z) term, relying only on σ(z)-y. When errors are large, weight updates are fast; when errors are small, weight updates are slow.

After determining the loss function, the model needs to select an optimization algorithm to iteratively solve for the optimal value. This paper selects the Adam optimization algorithm widely used in the deep learning field for iterative updates [25]. Finally, the overall model flow is shown in [Figure 5: see original paper]. This process is mainly divided into three parts: the text vectorization layer (embedding layer), the convolutional feature extraction layer, and the classification layer.

3 Experimental Validation

3.1 Data Preprocessing

To verify the accuracy of the microblog sentiment classification model constructed in this paper, comparative experimental validation was conducted. Taking the "Weibo Hot Search Rectification" topic on Sina Weibo as an example, this study used Python 3.6 to collect microblog user comment data through web crawling. The obtained data fields included user ID, nickname, personal profile, microblog content, mobile terminal type, repost/comment count, like count, and comment time. According to Baidu Index statistics, the lifecycle of the keyword "Weibo Hot Search Rectification" public opinion was from January 26, 2018, to February 28, 2018, as shown in [Figure 6: see original paper]. The data peak occurred on January 27, so this paper selects the first peak of this network public opinion as the starting point for sentiment tendency analysis, with February 28 as the endpoint. Microblog user comment information was collected under the "Weibo Administrator" user.

The original microblog comment data contained commercial advertisements, evaluation activities, and other noise data, which were not the required sentiment text data. To eliminate the impact of this noise data on experimental results, data cleaning was required before Chinese word segmentation. The data cleaning method used in this paper was a keyword-based identification method. Through analysis, it was concluded that most of this noise data contained the same main keywords, so comment data containing such keywords was removed, including comments that only "@ other microblog users" and anchor links starting with "http," forming a preliminary dataset totaling 134,765 entries. After initial screening, 116,764 entries remained. College students were recruited to manually classify them into positive and negative sentiment categories, with a total of 175 college students recruited. The voting method was used for manual binary classification of the initially screened 116,764 entries. After statistical classification results, samples with a vote difference greater than 25 were considered valid data to ensure obvious tendencies in positive and negative samples. For example, if a sample was a positive sample, it needed to receive at least 100 votes for positive samples and at most 75 votes for negative samples. The vote difference threshold was set to screen out ambiguous samples from the overall sample pool and ensure obvious sample tendencies, thereby facilitating model training. After screening, 91,376 entries were finally determined, including 38,185 positive sentiment entries and 53,191 negative sentiment entries.

3.2 Experimental Design

This paper designed three experiments:

First, to verify the improvement effect of introducing character-level vectors on the model, two groups of character vectors and word vectors with the same dimensions (128 dimensions) were trained using the same corpus data. A subset of 10,000 entries from the dataset was selected, with 5,000 positive and 5,000 negative samples each. Two convolutional neural networks with the same shape and structure were built, with convolution kernel shapes of 3$×$128, three such convolution kernels, combined with pooling layers of the same shape, and a final fully connected layer with a single neuron. Logistic regression and Gaussian kernel SVM classifiers were selected as two additional models for training and observation.

Second, to determine model hyperparameters and further verify the effectiveness and superiority of the multi-scale convolutional neural network microblog comment sentiment classification model based on character-word vectors, the experimental content and process were designed as follows: Training/test set division: 80% of the dataset was randomly selected as the training set, and the remaining 20% as the test set. The test set did not participate in training and was only used to verify model performance. Text vector training and selection:

To ensure text vector quality, the Sogou full-network news data corpus was used to train word vectors. This corpus collected over 20,000 real news articles from several domestic news sites during June-July 2012, covering 18 channels including society, sports, news, and entertainment. The data is authentic and covers a wide range of fields. Using this corpus for word vector training can minimize the specificity of word vectors and enhance generalization ability. Convolutional neural network model construction: The Keras deep learning toolkit was used with TensorFlow as its backend support. Hyperparameter tuning: To improve model classification performance, convolution kernel size, activation function, dropout random deactivation ratio, and number of iterations were adjusted. Effectiveness verification: Finally, the model's effectiveness was verified.

Third, to verify the model's ability to handle OOV words, the "Top 10 Internet Terms of 2017" released by the National Language Resources Monitoring and Research Center [26] were selected as OOV words. The number of relevant comment information containing these OOV words in the dataset was counted, and the network term with the largest number was selected to build a dataset. The classification accuracy of different models on this dataset was calculated to verify the model's ability to handle OOV words.

4 Discussion and Analysis

4.1 Chinese Character Vector Classification Ability Experiment

Using the same corpus data, character vectors and word vectors with the same dimensions (128 dimensions) were trained, with two groups each. A subset of 10,000 entries from the dataset was selected, with 5,000 positive and 5,000 negative samples each. Two convolutional neural networks with identical shape and structure were built, with convolution kernel shapes of 3$×$128, three such convolution kernels, combined with pooling layers of the same shape, and a final fully connected layer with a single neuron. Logistic regression and Gaussian kernel SVM classifiers were selected as two additional models for training and observation. The results are shown in .

** Chinese Character Vector Classification Ability**

Vector Type	CNN Accuracy	SVM Accuracy	Logistic Accuracy
Word Vector 1	0.7611	0.7313	0.7234
Word Vector 2	0.7802	0.7417	0.7501
Character Vector 1	0.8094	0.7231	0.7619
Character Vector 2	0.8010	0.7032	0.7242

Experimental results show that Chinese character vectors, combined with feature extraction by convolutional neural networks, can bring certain improvements in classification ability. At the same time, Chinese character vectors also demonstrate good classification effects in other models.

4.2 Hyperparameter Determination and Comparison Experiments for Multi-Scale Convolutional Neural Networks Based on Character-Word Vectors

After word vector training, hyperparameter tuning is needed to determine the optimal classification model. Hyperparameters differ from the model's own weight parameters and cannot be optimized through gradient descent. The selection of these hyperparameters largely affects the overall model performance. This paper focuses on convolution kernel size, activation function, dropout random deactivation ratio, and number of iterations as hyperparameters to be adjusted, discussing and analyzing them through experiments to determine the optimal hyperparameter combination.

4.2.1 Convolution Kernel Size Convolution kernel size is an important parameter in convolutional neural networks. Since one-dimensional convolution kernels are used in sentiment classification problems, only the length of the convolution kernel needs to be verified for its impact on model performance. The length of the convolution kernel determines how many neighboring words the sliding window will focus on [27]. This paper considers 2-6 adjacent words, with five types of convolution kernels. Combined with a three-layer convolution network architecture, six combination methods were designed for experimental comparison, as shown in .

** Experimental Comparison of Different Convolution Kernel Sizes**

Convolution Kernel Sizes (Layers)	Training Set Accuracy	Test Set Accuracy
2, 3, 4	0.9817	0.9422
4, 3, 2	0.8720	0.9129
3, 4, 5	0.9436	0.9712
5, 4, 3	0.9130	0.8776
4, 5, 6	0.9770	0.9418
6, 5, 4	0.8756	0.9128

Experimental results show that when the convolution kernel sizes of each layer are set to 3, 4, 5, the model achieves the highest classification accuracy on the test set, obtaining a model with higher classification precision. When set to 2, 3, 4, the model achieves the highest recall and F1 values, obtaining a model with higher positive example recognition rate. Meanwhile, when convolution kernel sizes are greater than or equal to 5, the model experiences overfitting, with a significant decrease in accuracy on the test set. The experimental results align with subjective understanding of language: when the distance between words exceeds five, their interaction often almost ceases to exist, making it impossible to extract effective semantic features using convolution kernels of this size.

4.2.2 Activation Function The activation function is an important component of the nonlinear transformation in neural networks. Common activation functions in today's deep neural networks are generally tanh and ReLU [28], with calculation methods shown in formulas (5) and (6):

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad \text{formula (5)}$$

$$ReLU(x) = max(0, x) \quad \text{formula (6)}$$

This paper uses the optimal convolution kernel size on the test set to adjust the activation functions of the convolutional and fully connected layers, as shown in .

** Experimental Comparison Results of Different Activation Functions on Test Set**

Activation Functions (Conv/FC)	Accuracy	Precision	Recall	F1 Value
tanh/tanh	0.8678	0.8942	0.9233	0.8715
tanh/ReLU	0.9034	0.9324	0.9272	0.9124
ReLU/tanh	0.8982	0.9233	0.9272	0.8715
ReLU/ReLU	0.7619	0.7856	0.8234	0.7432

Experimental results show that when the tanh function is used in the convolutional layer and the ReLU function is used in the fully connected layer, the model demonstrates optimal comprehensive performance. If ReLU is selected for the convolutional layer, it inherently destroys the initialized weights in the neural network. If the fully connected layer activation function is then selected as tanh, after several iterations, the input of the hyperbolic tangent function in the forward propagation process will concentrate on the positive half of the x-axis. The gradient value of the hyperbolic tangent function at infinity is zero, which will cause severe gradient vanishing problems.

4.2.3 Dropout Random Deactivation Ratio and Number of Iterations The random deactivation ratio of the Dropout layer in the network structure and the number of training iterations are also important factors affecting model performance. The following is the verification process for related experiments. First, the impact of the Dropout random deactivation ratio on model performance was tested, with relevant indicators on the test set shown in [Figure 7: see original paper]. The trend of the impact of the Dropout random deactivation ratio on the model can be observed. When the Dropout random deactivation ratio is between 0.05-0.1, the model's 各项指标 achieve optimal effect. When the random deactivation ratio exceeds 0.1, the overall indicators of the model show a downward trend. Therefore, setting the Dropout random deactivation ratio between 0.05-0.1 yields the best model performance.

The number of training iterations in deep neural networks requires manual adjustment of hyperparameters. The setting of the number of iterations often varies depending on the task being processed. Too few iterations prevent the model from converging to a local minimum, leading to underfitting. Conversely, too many iterations prolong model training time and also cause the model to face overfitting problems, losing its generalization ability. [Figure 8: see original paper] shows the changes in model-related indicators when the number of iterations is 1-20. From [Figure 8: see original paper], it can be seen that after 15 iterations, the model's 各项指标 have reached optimal effect. After 16 iterations, the 各项指标 show a clear downward trend, indicating that the model has begun to overfit. Therefore, this paper sets the number of iterations to 15.

4.2.4 Experimental Result Comparison To verify the superiority of the model constructed in this paper, mainstream sentiment classification models were selected for experimental result comparison on the same dataset. The experimental comparison results are shown in .

** Comparison Experimental Results of Sentiment Classification Models**

Model	Accuracy	Precision	Recall	F1 Value
CBOW+SVM	0.7844	0.7724	0.7738	0.7738
Word2Vec+CNN	0.8516	0.8499	0.8501	0.8501
Att+CNN	0.9029	0.8918	0.8749	0.8749
Multi-CNN	0.9473	0.9237	0.9274	0.9274

Among them, CBOW+SVM directly inputs text data through the N-gram model into SVM for classification; Word2Vec+CNN uses word vectors trained by Word2Vec and trains them using convolutional neural networks; Att+CNN uses an attention mechanism-based model with LSTM as the basic architecture to train the text matrix composed of word vectors. The comparative experimental data results show that the Multi-CNN model proposed in this paper has relatively high 各项指标 in short text classification tasks such as microblog comments and demonstrates relatively accurate classification effects.

4.3 Multi-Scale Convolutional Neural Network Based on Character-Word Vectors' Ability to Handle Out-of-Vocabulary Words

To verify the model's ability to handle OOV words, the "Top 10 Internet Terms of 2017" released by the National Language Resources Monitoring and Research Center were selected as OOV words, as shown in .

** Top 10 Internet Terms of 2017**

Term
打 call
你的良心不会痛么
惊不惊喜，意不意外
皮皮虾，我们走
扎心了，老铁
还有这种操作
你有 freestyle 么
…

The number of relevant comment information containing these OOV words in the dataset was counted. The network term "还有这种操作" had the largest number, reaching 1,146 entries, including 487 positive sentiment entries and 659 negative sentiment entries. The classification accuracy of different models on this dataset was calculated, with results shown in .

** Comparison of Sentiment Classification Model Accuracy**

Model	Accuracy
CBOW+SVM	0.534
Word2Vec+CNN	0.751
Att+CNN	0.769
Multi-CNN	0.873

It can be seen that due to the existence of OOV words, N-gram-based models like CBOW have difficulty constructing effective semantic features, resulting in poor classification effects. CNN-based models, due to the existence of convolutional layers, can effectively perform feature extraction, and their classification effects are far superior to CBOW. The Multi-CNN model, which uses more fine-grained character-level vectors, has a significantly higher accuracy than other CNN models, thus proving its good ability to handle OOV words.

5 Research Conclusions

At the theoretical level, this paper proposes a sentiment classification model for microblog comments based on multi-scale convolutional neural networks with character-word vectors, addressing existing challenges in microblog comment sentiment classification such as limited contextual information and numerous OOV words. The model uses convolutional layers to learn structural and semantic features in microblog comments and employs multi-scale convolution kernels to capture connections between words of different degrees, effectively focusing on contextual information at different levels, thereby improving the constraints of limited contextual information in microblog comments. The combination of character and word vectors alleviates the impact of OOV words on model sentiment classification performance. By crawling real microblog comment data, the feasibility of the model was verified. This model can provide guidance for relevant public opinion supervision departments to correctly manage the development of topic events and the direction of public opinion, offering more feasible management methods for the dynamic regulation and monitoring of Weibo public opinion.

In practical terms, this paper provides ideas for setting hyperparameters through experimental methods and compares mainstream sentiment classification research methods to construct and verify the microblog comment sentiment classification model. By using convolutional neural networks, it is possible to analyze the degree of attention and emotional changes of network users toward topic events, thereby providing guidance for relevant public opinion supervision departments to correctly manage the development of topic events and public opinion, offering more feasible management methods for the dynamic regulation and monitoring of Weibo public opinion.

This paper uses the combination of character-word vectors and multi-scale convolution kernels to solve OOV word problems to a certain extent, but there are still certain limitations in the research process. Convolutional neural network training takes a long time and is very sensitive to hyperparameter settings. In subsequent research, regularization methods will be considered to reduce the risk of model overfitting and training difficulty, thereby further improving the model's generalization ability.

References

[1] Sina Technology. Weibo Releases Unaudited Financial Results for the Second Quarter of 2018 [EB/OL]. [2018-10-08]. https://tech.sina.com.cn/i/2018-08-08/doc-iihkukt9159883.shtml.

[2] NEVES SILVAR, GAMITO M, PINA P, et al. Modelling influence and reach in sentiment analysis [J]. Procedia CIRP, 2016, 47: 48-53.

[3] SOLEYMANI M, GARCIA D, JOU B, et al. A Survey of Multimodal Sentiment Analysis [J]. Image & vision computing, 2017, 65(9): 3-14.

[4] ÖZTÜRK N, AYVAZ S. Sentiment analysis on Twitter: a text mining approach to the Syrian refugee crisis [J]. Telematics & informatics, 2018, 35(4): 136-147.

[5] Tang Xiaobo, Lan Yuting. Sentiment Analysis of Weibo Product Reviews Based on Feature Ontology [J]. Library and Information Service, 2016(16): 121-127.

[6] Ma Songyue, Xu Xin. Research on User Online Evaluation Based on Comment Sentiment Analysis—Taking Douban Movies as an Example [J]. Library and Information Service, 2016(10): 95-102.

[7] Liang Xiaomin, Xu Jian. Research on Sentiment Analysis of Comment Objects and Their Relationship Networks in Public Opinion Events [J]. Information Science, 2018, V36(2): 37-42.

[8] Tang Huifeng, Tan Songbo, Cheng Xueqi. Comparative Study of Chinese Sentiment Classification Techniques Based on Supervised Learning [J]. Journal of Chinese Information Processing, 2007, 21(6): 88-94.

[9] Zhang Dongwen, Yang Pengfei, Xu Yunfeng. Research on Chinese Comment Sentiment Classification Based on Word2vec and SVMperf [J]. Computer Science, 2016, 43(6): 418-421.

[10] NIHAR M. R, RAJESH S. P. LFNN: Lion Fuzzy Neural Network based evolutionary model for text classification using context and sense based features [J]. Applied soft computing, 2018, 71: 994-1008.

[11] Feng Xingjie, Zhang Zhiwei, Shi Jinchuan. Text Sentiment Analysis Based on Convolutional Neural Network and Attention Model [J]. Application Research of Computers, 2018, 35(5): 1434-1436.

[12] ZHANG X, ZHAO J, LECUN Y. Character-level convolutional networks for text classification [J]. Computer science, 2015(9): 1.

[13] Pang Jun. Research on Blog Text Clustering Based on Determined Topics and Sentiment Polarity [D]. Wuhan: Wuhan University of Technology, 2010.

[14] Wang Xiwei, Zhang Liu, Wen Qing. Research on User Sentiment Evolution of Network Public Opinion in Mobile Environment Based on Bayesian Model—Taking the Topic of "Chinese Women's Volleyball Team Winning the Rio Olympics" on Sina Weibo as an Example [J]. Journal of the China Society for Scientific and Technical Information, 2018, 37(12): 1241-1248.

[15] ZHANG Q, GAO Z, LIU J. Research of Weibo short text classification based on Word2vec [J]. Netinfo security, 2017(1): 57-63.

[16] Zhou Ying, Liu Yue, Cai Jun. Microblog Sentiment Analysis Based on Attention Mechanism [J]. Information Theory and Practice, 2018, 41(3): 89-94.

[17] Kong Xing, Lin Qing. Review of Sentiment Classification Research on Subjective Texts [J]. Information Technology, 2018(8): 126-130.

[18] Word2vec. [EB/OL]. [2018-10-08]. https://en.wikipedia.org/wiki/Word2vec.

[19] Zhou Lian. Exploration of Word2vec Working Principle and Application [J]. Library and Information Guide, 2015(2): 145-148.

[20] Lu Hongtao, Zhang Qinchuan. Research on Application of Deep Convolutional Neural Network in Computer Vision [J]. Journal of Data Acquisition and Processing, 2016, 31(1): 1-17.

[21] Wu Peng, Liu Hengwang, Shen Si. Research on Network Public Opinion Sentiment Recognition Based on Deep Learning and OCC Emotion Rules [J].

Journal of the China Society for Scientific and Technical Information, 2017, 36(9): 972-980.

[22] REN Y, JI D, REN H. Context-augmented convolutional neural networks for twitter sarcasm detection [J]. Neurocomputing, 2018, 308(9): 1-7.

[23] Yu Zheng. Research and Application of Text Vectorization Based on Deep Learning [D]. Shanghai: East China Normal University, 2016.

[24] Wang Yi, Xie Juan, Cheng Ying. Deep Neural Network Language Model Combining LSTM and CNN Hybrid Architecture [J]. Journal of the China Society for Scientific and Technical Information, 2018(2): 194-205.

[25] SMITH L N. Cyclical learning rates for training neural networks [J]. Computer science, 2017: 464-472.

[26] Top 10 Internet Terms of 2017. [EB/OL]. [2019-02-22]. https://baike.baidu.com/item/2017年度十大网络用语/22273910?fr=aladdin.

[27] Chen Ke, Liang Bin, Ke Wende, et al. Chinese Microblog Sentiment Analysis Based on Multi-Channel Convolutional Neural Network [J]. Journal of Computer Research and Development, 2018, 55(5): 945-957.

[28] Qu Zhilin, Hu Xiaofei. Research on Convolutional Neural Network Based on Improved Activation Function [J]. Computer Technology and Development, 2017, 27(12): 77-80.

Author Contributions:
Zhang Liu: Responsible for paper writing, revision, data collection, and data processing;
Wang Xiwei: Responsible for research proposition and research framework formulation, paper writing and final revision;
Huang Bo: Responsible for data processing and technical support;
Liu Yutong: Responsible for data processing and abstract translation.

Abstract: [Purpose/Significance] The microblog commentary sentiment classification model can play a guiding role for relevant public opinion supervision departments to correctly control the development of topic events and public opinion. [Method/Process] Based on the multi-scale convolutional neural network of word vector, this paper used multivariate convolution kernels to improve the conditional constraints of finite context information in microblog commentary, and constructed a microblog commentary emotion classification model based on multi-scale convolutional neural network of word vector. Finally, the feasibility and superiority of the model were verified by crawling the real data of "microblogging hot search and rectification". [Result/Conclusion] Verification results show that the multi-scale convolutional neural network based on word vector performs well in the short text classification task with limited context information such as Weibo public opinion. On the theoretical level, this paper provides a more accurate emotional classification theoretical model and classification method for Weibo public opinion emotional classification. In practice, it

can better guide the public opinion supervision department to better guide and supervise the emotional tendency of public opinion.

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