symmetry

MDPI

*Article*

# Short Text Classification Based on Explicit and Implicit Multiscale Weighted Semantic Information

**Jun Gong [1], Juling Zhang [2,*], Wenqiang Guo [2], Zhilong Ma [2] and Xiaoyi Lv [3]**

[1] Teaching Affairs Division, Xinjiang University, Urumqi 830046, China; xjuzyk@xju.edu.cn
[2] The School of Cyberspace Security, Xinjiang University of Finance and Economics, Urumqi 830012, China; gwq@xjufe.edu.cn (W.G.); mzl@xjufe.edu.cn (Z.M.)
[3] The School of Software, Xinjiang University, Urumqi 830046, China; lvxiaoyi@xju.edu.cn
* Correspondence: zhangjuling@xjufe.edu.cn; Tel.: +86-151-9917-6691

**Abstract:** Considering the poor effect of short text classification due to insufficient semantic information mining in the current short text matching methods, a new short text classification method is proposed based on explicit and implicit multiscale weighting semantic information interaction. First, the explicit and implicit representations of short text are obtained by a word vector model (word2vec), convolutional neural networks (CNNs), and long short-term memory (LSTM). Then, a multiscale convolutional neural network obtains the explicit and implicit multiscale weighting semantics information of short text. Finally, the multiscale weighting semantics is fused for more accurate short text classification. The experimental results show that this method is superior to the existing classical short text classification algorithms and two advanced short text classification models on the five short text classification datasets of MR, Subj, TREC, SST1 and SST2 with accuracies of 85.7%, 96.9%, 98.1%, 53.4% and 91.8%, respectively.

**Keywords:** convolution neural network; explicit semantics; implicit semantics; recurrent neural network; short text classification



**Citation:** Gong, J.; Zhang, J.; Guo, W.; Ma, Z.; Lv, X. Short Text Classification Based on Explicit and Implicit Multiscale Weighted Semantic Information. *Symmetry* **2023**, *15*, 2008. https://doi.org/10.3390/sym15112008

Academic Editor: Theodore E. Simos

Received: 26 September 2023
Revised: 22 October 2023
Accepted: 23 October 2023
Published: 1 November 2023



## 1. Introduction

With the rapid development of big data and mobile Internet technology, network data have greatly penetrated people's lives. Weibo, Facebook, WeChat, forums, TikTok and other application websites and software have become important methods for people to record their lives and express their feelings, resulting in considerable data information flow, in which short text data account for a large proportion. Considering these massive short text data, one method for obtaining valuable information quickly and accurately is to classify short texts [1]. It can help all kinds of people, businesses and institutions to obtain the useful data they need in a targeted manner. Thus, the value utilization rate of these short text data is improved.

To classify short texts, how to express short texts must be considered. Generally, the character form of the original short texts is converted into its corresponding real vector form. Because the short text is generally less than 100 words, the short length makes the short text contain fewer features, so the converted data have the characteristics of high dimension and sparseness. In traditional short text classification, the one-hot method [2] or constructing a text vector space model [3] is often used to represent text. These methods are simple and effective but also easily cause dimensional disasters [4]. With the rise and application of deep learning, scholars have studied using deep learning to obtain text representation. In 2013, Mikolov proposed the word embedding method [5], and the word vectors obtained not only describe the word information from the semantic level but also avoid dimension disaster risk. Short texts use the word vector as the deep learning model input, greatly improving text classification performance. There are many text classification algorithms in natural language processing (NLP). The main method is

deep learning, and the typical representatives are CNNs [6] and recurrent neural networks (RNNs) [7]. Kim [8] transformed character text into a real matrix similar to an image by a word vector method and directly applied CNNs to text classification. Arevian et al. directly applied RNNs to text classification and demonstrated that it is effective for context-sensitive email classification [9]. Kalchbrenner et al. improved the pooling layer in CNNs. They used dynamic k-max pooling to replace the original max pooling mechanism [10]. Yin Yu extracted the feature of music information based on SVM, CNN and RNN, and constructed a CNN-LSTM model to classify music emotion [11]. Tang et al. proposed an RNN structure for series-parallel gate valves to handle short text feature representation and classification [12]. Zhou et al. serially processed CNN and LSTM, and used the high-dimensional semantic representation extracted by the CNN as the LSTM text classification input [13]. Reference [14] studied the LSTM network by the differential evolution algorithm (DEA). Governmental organizations and healthcare facilities utilized machine learning and deep learning algorithms to anticipate the daily incidence of COVID-19 for India over the course of the next six months, and famous timeseries algorithms were implemented including LSTM, Bi-Directional LSTM and Stacked LSTM and Prophet [15].

In the CNN and RNN methods, the CNN only considers the local semantic features and ignores the global semantic features of the text. The RNN does not fully consider the local semantic features of the text, so both methods have insufficient feature mining.

To fully mine the semantic information of short texts, both explicit and implicit semantic information of local and global texts are used, and a short text classification method based on explicit and implicit multiscale weight semantic information is proposed to address the existing problems in the above two mainstream methods.

(1) Obtaining explicit and implicit multiscale semantic information matrices of short texts by fine-tuning the word embedding model, CNN and Bi-LSTM;

(2) Mining high-order characteristic semantic information of multiscale explicit and implicit semantic information through weight convolution;

(3) The explicit and implicit multiscale weight feature information is fused after mining.

The method proposed in this paper can fully mine and use the local and global explicit and implicit semantic information of short texts, which is more effective for short text classification. Experiments on five public datasets fully verify that the method proposed in this paper is more effective.

## 2. Acquisition of Explicit and Implicit Multiscale Semantic Information

### *2.1. The Process of Explicit and Implicit Multiscale Semantic Information*

To fully mine the semantic information of short text data, this paper constructs explicit and implicit multiscale weight semantic information for short text data. It classifies short texts by mining explicit and implicit multiscale weight semantic information.

Obtaining explicit and implicit multiscale semantic information from short texts is shown in Figure 1.

### *2.2. Acquisition of Explicit Multiscale Semantic Information*

The short text is first input and converted into a word vector to obtain the semantic representation of the short text, which contains the semantic expression of the word. The short corpus is used to train the word vector model, skip-gram, to make the word vector more suitable for the short sentence characteristics. Because the skip-gram model does not make full use of word order information, the coincidence between the trained vector and the short text is not high enough. Therefore, this paper proposes incorporating word order information into the word vector input model to help the model learn better word vector representation.

In this paper, the pretrained word vector is a 300-dimensional word vector from Google, which is used as the initialization vector of the word vector. The short text dataset to be classified is used as the corpus. First, the word position of each word in the short text

is vectorized and then added to the initialization word vector. Then, a new word vector that is more in line with the short text characteristics is microtrained in the skip-gram model.

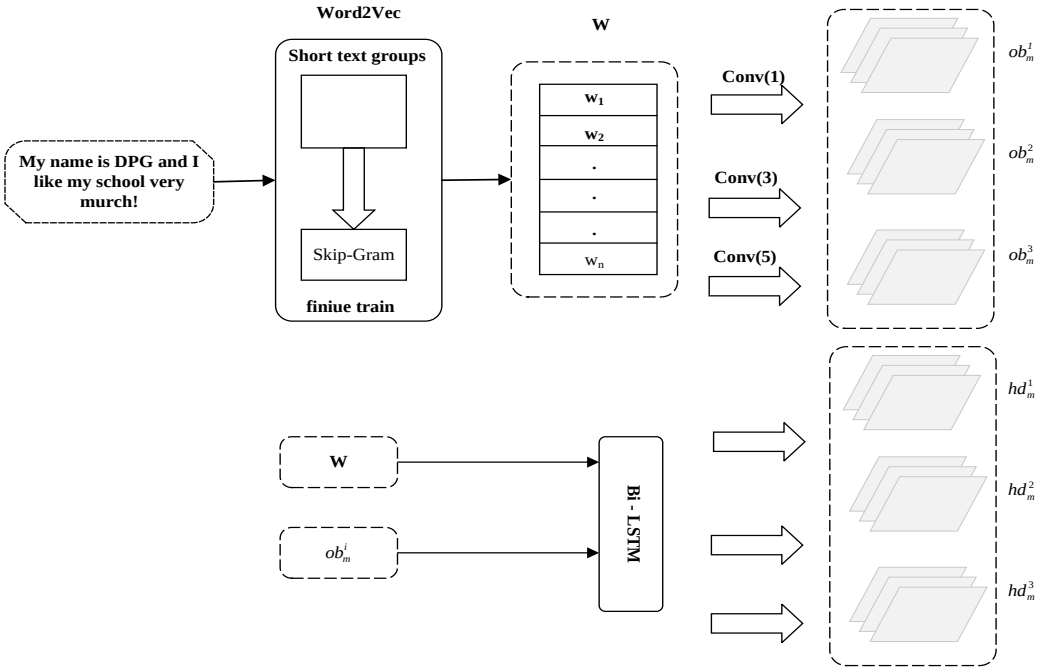


**Figure 1.** The acquisition process of explicit and implicit multiscale semantic information.

The skip-gram model is a commonly used unsupervised model for training word vectors. The basic idea is to use current words to predict words in context. The selected predicted words are symmetrical with respect to the current words, that is, if $p$ words are selected above, then $p$ words are also selected below. In this paper, $p = 2$. If $w_i$ is the current word input by the model, then $w_{i-2}$, $w_{i-1}$, $w_{i+1}$ and $w_{i+2}$ are the predicted contexts, where the width of the context is 2. Formula (1) is used to calculate the conditional probability of words in context.

$$P(w_{i-2}, w_{i-1}, w_{i+1}, w_{i+2}|w_i) = P(w_{i-2}|w_i) * P(w_{i-1}|w_i) * P(w_{i+1}|w_i) * P(w_{i+2}|w_i). \quad (1)$$

In constructing explicit multiscale semantic information, the short text is first segmented to obtain the corresponding word sequence $S_1 = [w_1, w_2, \cdots, w_n]$. Then, the word vector matrix $W \in R^{n \times d}$ of the short text shown in Figure 1 is obtained by word vectorization through the microtrained word vector model, where $n$ is the number of words and $d$ is the word vector dimension. To fully capture the multiscale semantic information of short texts, the multiscale convolution network is used to convolve $W$, the convolution kernel sizes are 1, 3 and 5, and the explicit multiscale semantic information matrix $ob_m^1, ob_m^2, ob_m^3 \in R^{n \times d}$ is obtained by convolving $W$.

### *2.3. Acquisition of Implicit Multiscale Semantic Information*

The Bi-LSTM obtains implicit semantic information, and the bidirectional hidden state splicing in Bi-LSTM is taken as the implicit semantic information at that moment. Notably, the word vector needs to be preprocessed before it is transmitted to Bi-LSTM, which is preprocessed by Formulas (2)–(4).

$$hd_m^{1i} = concate(\overrightarrow{h_l}, \overleftarrow{h_l}) \quad (2)$$

$$\overrightarrow{h_l} = \overrightarrow{lstm}(\overrightarrow{W_l}|ob_m^{li}) \quad (3)$$

$$\overleftarrow{h_l} = \overleftarrow{lstm}(\overleftarrow{W_l}\,|ob_m^{li}) \tag{4}$$

The acquisition process of implicit semantic information is as follows. First, the original word vector matrix $W$ and explicit semantic information $ob_m^1, ob_m^2, ob_m^3$ are spliced in the dimension. Then, the spliced vector sequence is transmitted to Bi-LSTM, and Bi-LSTM captures the implicit multiscale semantic information $hd_m^1, hd_m^2, hd_m^3 \in R^{n\times d}$ containing the context of short texts.

In this paper, the traditional RNN is used to improve the Bi-LSTM structure for mining the implicit semantic information of short texts, which alleviates the problems that the traditional RNN is not good at dealing with the gradient dispersion of sequence data and long-dependent sequence data. These are due to the RNN forget gate, input gate and output gate functions. The forget gate determines how much information of the cell state in the last moment remains in the current moment, and the input gate determines how much information input at the current moment is saved to the current cell state. The output gate determines how much information of the current cell state can be output to the output gate.

The workflow of these three gates is as follows: let $W_f, W_i, W_c, b_f, b_i, b_c$ and $b_o$ be the training parameters of the Bi-LSTM model, $x_i$ be the current input, $h_{i-1}$ be the last hidden state, $o_t$ be the current output, $h_t$ be the current hidden state, $f_t$ be the forgotten threshold, $i_t$ be the input threshold, $\tilde{C}_t$ be the temporary cell state, $C_t$ be the current cell state, and $\delta(*)$ and $tanh(*)$ be *sigmoid* and *tanh* functions, respectively. Use Formula (5) to calculate the information distribution of the forget gate, Formula (6) to calculate the information distribution of the input gate, Formula (7) to calculate the temporary cell state in the network, Formula (8) to calculate the network cell state at the current moment, Formula (9) to calculate the information distribution of the output gate, and Formula (10) to calculate the network hidden layer output information at the current moment.

$$f_t = \delta(W_f \cdot [h_{t-1}, X_t] + b_f) \tag{5}$$

$$i_t = \delta(W_i \cdot [h_{t-1}, X_t] + b_i) \tag{6}$$

$$\tilde{C}_t = tanh(W_c \cdot [h_{t-1}, X_t] + b_c) \tag{7}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{8}$$

$$O_t = \delta(W_o \cdot [h_{t-1}, X_t] + b_o) \tag{9}$$

$$h_t = o_t * tanh(C_t) \tag{10}$$

## 3. Semantic Information Mining and Text Classification

### *3.1. Explicit and Implicit Semantic Information Interaction and Feature Mining*

The explicit multiscale semantic information $ob_m^1, ob_m^2, ob_m^3 \in R^{n\times d}$ and implicit multiscale semantic information $hd_m^1, hd_m^2, hd_m^3 \in R^{n\times d}$ of the short text have been obtained. Now, the weight convolution operation is carried out. The mining process of explicit and implicit multiscale semantic information is shown in Figure 2.

First, the explicit and implicit multiscale semantic information are multiplied according to the corresponding scales to obtain the corresponding weight interaction matrix. For example, the explicit scale semantic information $ob_m^1$ and the corresponding implicit scale semantic information $hd_m^1$ are multiplied to obtain the weight interaction matrix $oh_m^1 \in R^{n\times n}$ under this scale. To further obtain the weight matrix corresponding to explicit and implicit scale semantic information, two importance coefficients $ob_m^i \in R^n$ and $hd_m^i \in R^n$ with length $n$ are obtained by adding the weight interaction matrix $oh_m^i$ in rows and columns,

respectively, which correspond to the importance of explicit and implicit scale semantics, respectively. Then, the weight coefficients $\alpha^i$ and $\beta^i$ are calculated by Formula (11), and the calculated weight coefficients are multiplied by the corresponding position and explicit and implicit scale semantic information by Formula (12) to obtain the sum of weight scale semantic information $wob_m^i \in R^{n\times d}$ and $whd_m^i \in R^{n\times d}$.

$$\alpha^i = \beta^i = \frac{e^{x_i}}{\sum_1^n e^{x_j}}, x \in R^{n\times d} \tag{11}$$

$$wob_m^i = ob_m^i * \alpha^i, whd_m^i = hd_m^i * \beta^i \tag{12}$$

Similarly, according to the above method, the explicit multiscale weighted semantic information $wob_m^1, wob_m^2, wob_m^3 \in R^{n\times d}$, implicit multiscale weighted semantic information $whd_m^1, whd_m^2, whd_m^3 \in R^{n\times d}$ and explicit and implicit multiscale semantic interactive information $oh_m^1, oh_m^2, oh_m^3 \in R^{n\times n}$ are obtained by weight convolution of explicit and implicit multiscale semantic information.

Finally, a CNN is used to extract high-order features of three kinds of semantic information matrices, and explicit features $ob_{vec}$, implicit features $hd_{vec}$ and explicit-implicit interactive features $oh_{vec}$ are obtained.

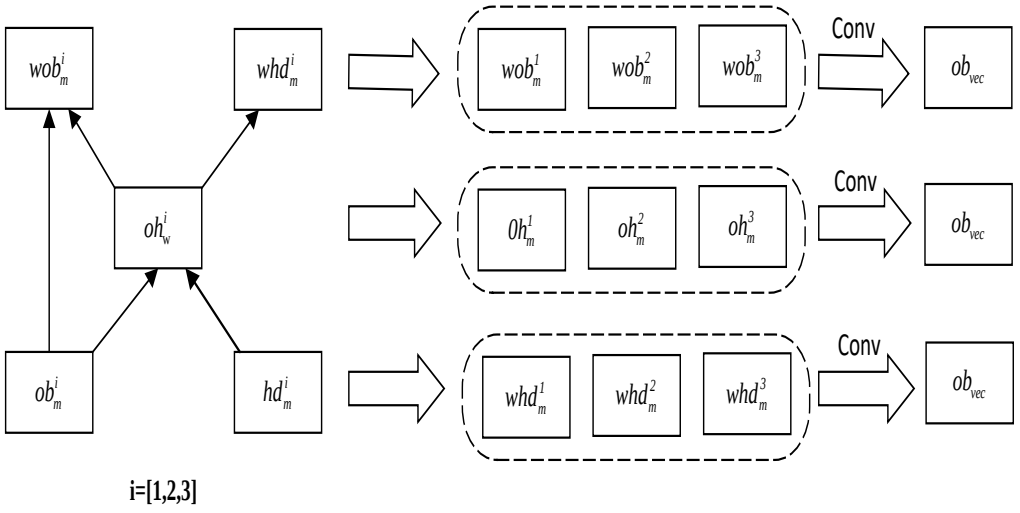


**Figure 2.** The mining process of explicit and implicit multiscale weight semantic information.

### *3.2. Semantic Feature Fusion and Text Classification*

Previously, the explicit feature $ob_{vec}$, implicit feature $hd_{vec}$ and explicit-implicit interactive feature $oh_{vec}$ of the short text were obtained. To better obtain the semantic expression of short texts, it is necessary to fuse these semantic features of short texts obtained by the final model classification.

The semantic feature $S_{vec}$ of short texts is obtained by Formulas (13)–(16), and finally, short texts are classified based on this semantic feature.

$$obhd_{vec} = [ob_{vec}; hd_{vec}; ob_{vec} * hd_{vec}] \tag{13}$$

$$oboh_{vec} = [ob_{vec}; oh_{vec}; ob_{vec} * oh_{vec}] \tag{14}$$

$$hdoh_{vec} = [hd_{vec}; oh_{vec}; hd_{vec} * oh_{vec}] \tag{15}$$

$$s_{vec} = obhd_{vec} + oboh_{vec} + hdoh_{vec} \tag{16}$$

## 4. The Experimental Result

### *4.1. Experimental Environment and Dataset*

To fully verify the effectiveness and stability of the proposed method, this paper experiments on five standard public datasets and compares it with the existing classical short text classification method and two advanced short text classification methods.

The experimental environment of this paper is the TensorFlow14.04 deep learning framework, Python3.6 development semantic environment and Ubuntu14.04 operating system. To speed up model training, an NVIDIA 1080 24 G GPU is used to speed up the operation during model training.

To fully verify the effectiveness of the short text classification method with explicit and implicit semantic information interaction proposed in this paper, the proposed method is trained and verified on five datasets, MR, Subj, TREC, SST1 and SST2, and its performance is compared with the existing algorithms. Table 1 shows the situation of each experimental dataset, where *c* is the number of categories in the dataset, *len* is the average sentence length of short texts, that is, the average number of words, *train* is the number of model training datasets, *dev* is the number of model verification datasets, *test* is the number of model testing datasets, $|V|$ is the dictionary size in the dataset, that is, the number of words, and $CV$ indicates that cross-validation is adopted (here, tenfold cross-validation is adopted).

**Table 1.** Dataset information.

| Data | *c* (pcs) | *len* (pcs) | *train* (pcs) | *dev* (pcs) | *test* (pcs) | $\mid V \mid$ |
|---|---|---|---|---|---|---|
| MR | 2 | 21 | 10,662 | - | CV | 20,191 |
| Subj | 2 | 23 | 10,000 | - | CV | 21,057 |
| TREC | 6 | 10 | 5452 | - | 500 | 9137 |
| SST1 | 5 | 18 | 8544 | 1101 | 2210 | 17,836 |
| SST2 | 2 | 19 | 6920 | 872 | 1821 | 16,185 |

MR is a binary short text dataset of positive and negative comments. Each sentence represents a user comment, composed of positive and negative comment data used to identify positive and negative user comments.

Subj is a two-category short text dataset about subjectivity and objectivity. It consists of 5000 subjective and objective sentences used to identify sentence subjectivity and objectivity.

TREC is a multiclassification short text dataset of six types of questions abbreviation, description, entity, personnel, place and numerical questions. TREC is used to identify sentence question types.

SST1 is a fine-grained extension of the MR dataset, which is no longer limited to positive and negative comments, but is refined into five emotional tendencies: very negative, negative, neutral, positive and very positive. The dataset is divided into three parts, 8544 training, 1101 verification and 2210 test datasets, to identify the fine-grained emotional tendencies of sentences.

SST2 is a variant of the SST1 dataset. SST2 combines very negative comments and negative comments into negative comments and very positive comments and positive comments into positive comments, removes neutral comments, and finally forms a dataset containing negative comments and positive comments, which is used to identify the negative and positive tendencies of sentences.

### *4.2. Experimental Model and Evaluation Criteria*

Because the number distribution of each category in the five datasets is relatively uniform, and it belongs to binary classification or multiclassification tasks, this experiment evaluates the performance accuracy, and the final dataset accuracy with tenfold cross-validation is the average of tenfold cross-validation accuracy.

To verify the advantages of the method proposed in this paper over other methods, it is compared with the following classical short text classification algorithms and two new short text classification algorithms.

FastText is a very simple and efficient fast text classification algorithm proposed by the Facebook AI Lab but with the same accuracy as the neural network method. Its goal is to simply add the word vectors of short texts to form a classification vector and then classify the text, often used by the word vector training model [16].

CNN-nonstatic, CNN-static and CNN-multichannel are typical text classification methods based on CNN [17], proposed by Kim in 2014, used CNN for short text classification based on image recognition and achieved good classification results. CNN-nonstatic uses the trained word vectors to initialize short text, and the word vectors are updated together during the training process. CNN-static uses the trained word vectors to initialize short text but does not update the word vectors during the training process. CNN-multichannel is similar to the principle of the image channel, and CNN-static and CNN-nonstatic are combined into a two-channel matrix for training.

MVCNN is a new convolutional network model. In 2016, Yin et al. proposed an algorithm to combine different versions of pretrained word vectors and use a multiscale convolution kernel to extract multigranularity semantic features of the text to be classified, thus improving text classification accuracy [18].

LSTM, Bi-LSTM and Tree-LSTM are RNN structures for processing time series datasets. LSTM, a long-term and short-term memory network, alleviates the gradient dispersion and weak long-term dependence of sequences in traditional RNNs, further improving the ability to process sequential data. Bi-LSTM is a bidirectional LSTM structure that enables the network at a certain moment to access the information of the past context simultaneously. Tree-LSTM applies the LSTM network structure to the tree structure network [19].

RCNN is a variation in the RNN. The forward output, backward output and word vector of the Bi-LSTM network are cascaded, and the classification vector is formed by a pooling operation for text classification [20].

SPCGRU is a short text classification based on a serial-parallel convolution gate valve [21].

BLSTM_MLPCNN is a method that combines character-level vectors and word vectors as the model input and combines the Bi-LSTM network and MLPCNN to mine text semantic feature vectors for short text classification [22].

### *4.3. Experimental Setup and Experimental Results*

Some parameters in the model training need to ensure the consistency of the experiment. Here, the word vector dimension is set to 300, the hidden layer unit dimension of LSTM is set to 300, and the short text length is set to 60. If the length is less than 60, it is filled with 0, and the part exceeding 60 is truncated. The batch size of the model training is set to 512, the initial learning rate is 0.01, and the learning rate is dynamically adjusted according to the exponential decay strategy [23]. Stochastic gradient descent (SGD) reduces the model parameters in a random gradient way [24].

Four methods are used to set the word vector of the model, static, nonstatic, prestatic and nonprestatic, in which static means using the pretrained Google 300-dimensional word vector for initialization and keeping it unchanged during model training; prestatic means that word vectors are initialized by pretraining Google's 300-dimensional word vectors, and short texts of five datasets are used as corpora for microtraining word vectors and remain unchanged during the training process; nonstatic indicates that the short text is initialized with the pretrained Google's 300-dimensional word vector and updated with the model parameters during the model training process;and nonprestatic means that the pretrained Google's 300-dimensional word vector and short text data of five datasets are used as corpora to be input to the skip-gram model for word vector microtraining, and updated with model parameters in model training.

In this paper, experiments are carried out on 5 data sets and compared with 11 other algorithms. The experimental results are shown in Table 2.

**Table 2.** Test results (accuracy %).

| Model | MR (%) | Subj (%) | TREC (%) | SST1 (%) | SST2 (%) |
|---|---|---|---|---|---|
| FastText | 79.7 | 91.5 | 92.1 | 46.1 | 85.4 |
| CNN-static | 81.0 | 93.0 | 92.8 | 45.5 | 86.8 |
| CNN-non-static | 81.5 | 93.4 | 93.6 | 48.0 | 87.2 |
| CNN-multi-channel | 81.1 | 93.2 | 92.2 | 47.4 | 88.1 |
| MVCNN | - | 93.9 | - | 49.6 | 89.4 |
| LSTM | - | - | - | 46.4 | 84.9 |
| Bi-LSTM | - | - | - | 49.1 | 87.5 |
| LSTM-Tree | - | - | - | 51.0 | 88.0 |
| RCNN | - | - | - | 47.2 | - |
| SPCGRU | 82.3 | 94.9 | 95.8 | - | - |
| BLSTM MLPCNN | 83.0 | 95.0 | 95.7 | 49.0 | 88.2 |
| Ours-static | 84.1 | 95.9 | 97.3 | 52.1 | 90.6 |
| Ours-prestatic | 84.4 | 95.8 | 96.9 | 52.4 | 89.9 |
| Ours-nonstatic | 85.1 | 96.3 | 97.6 | 52.7 | 90.8 |
| Ours-prenonstatic | 85.7 | 96.9 | 98.1 | 53.4 | 91.8 |

In Table 2, the short text classification method based on explicit and implicit semantic information interaction has a classification accuracy of 85.7% on MR data sets, which is 2.7% higher than that of all the compared models. The classification accuracy on the Subj data set is 96.9%, 1.9% higher than that in the contrast model. The classification accuracy on the TREC data set is 98.1%, which is 2.4% higher than that in the contrast model. The classification accuracy on the SST1 dataset is 53.4%, 2.4% higher than that in the contrast model. The classification accuracy on the SST2 dataset is 91.8%, 2.7% higher than the best result in the compared model.

According to the classification results on five datasets, the short text classification method based on explicit and implicit semantic information interaction proposed in this paper performs best on MR, Subj, TREC, SST1 and SST2, and the performance on each data set improved. Therefore, the short text classification method based on explicit and implicit multiscale weight semantic information proposed in this paper is very effective.

## 5. Analysis of Experimental Results

From Table 2, it is evident that our proposed approach outperforms other text classification methods across the five datasets. Notably, in the non-pre-static setting, our classification accuracy exceeds that of alternative methods by over 2% for all five datasets. In conducting comparative experiments between Our-static and Our-pre-static, as well as Our-non-static and Our-pre-non-static, we observed that incorporating word vector micro-training results in superior model performance. This suggests that word vectors subjected to micro-training are better aligned with the target dataset. Additionally, when comparing Our-static and Our-non-static, and Our-pre-static and Our-pre-non-static, we observed that fine-tuning (non-static) word vectors with training tasks yields better results than using a static word vector. This demonstrates that training word vectors with specific tasks produces semantics that are more contextually relevant to the target data.

The effect of Ours-prenonstatic on the MR dataset, Subj dataset, TREC dataset, SST1 dataset and SST2 dataset is better than that of Ours-static, Ours-prestatic and Ours-nonstatic, which shows that microtraining word vectors and dynamically updating word vectors can further improve the classification performance of the model.

To analyse the stability of the short text classification method based on explicit and implicit semantic information interaction proposed in this paper, a stability analysis experiment of the model is carried out, and the results are shown in Table 3.

**Table 3.** Model stability experiment (accuracy %).

| $\beta$ | MR (%) | Subj (%) | TREC (%) | SST1 (%) | SST2 (%) |
|---|---|---|---|---|---|
| 0.5 | 85.7 | 96.8 | 98.0 | 53.4 | 91.7 |
| 0.7 | 85.6 | 96.8 | 98.0 | 53.3 | 91.6 |
| 1.0 | 85.7 | 96.7 | 98.1 | 53.4 | 91.8 |
| 1.3 | 85.7 | 96.9 | 98.1 | 53.2 | 91.5 |
| 1.5 | 85.6 | 96.7 | 98.1 | 53.2 | 91.5 |

The superparameter $\beta$ value in the model is set to [0.5, 0.7, 1.0, 1.3, 1.5]. Here, we set the superparameter $\beta$ to different values to illustrate the stability of our model. By observing the fluctuation of the model on various datasets, it can be found that the accuracy of the model fluctuates by 0.1% in the MR dataset, by 0.2% in the Subj dataset; by 0.1% in the TREC database, by 0.2% in the SST1 dataset and by 0.3% in the SST2 dataset. We can see our results are not fluctuate drastically with the change of superparameter $\beta$. This also illustrates that our model is stable.

## 6. Conclusions

Aiming at the short and sparse features of short text data sentences, this paper designs a short text classification method based on explicit and implicit multiscale weight semantic information. This method uses explicit multiscale semantic information and implicit multiscale semantic information of short text data simultaneously, and carries out weight convolution and interaction between the semantic information, further mining the deep semantic features of short text on the interactive information, thus enriching the classification semantic information of short text classification. Comparative experiments on five short text datasets also show that the proposed method can mine more semantic information from the original short text data, and provide more classification feature information for the model to improve its text classification ability. This method can be used for public opinion analysis and early warning of current college students, academic situation analysis, teaching effect analysis, etc. With the continuous development of deep learning, there are many models with better performance that can better extract feature information, such as EfficientNet and VIT models in the CV field and Bert and XL-Net models in the NLP field. Therefore, in future work, we will try to use these more advanced model frameworks to extract the explicit and implicit features of data and explore other methods in feature fusion to obtain better model effect.

**Author Contributions:** J.G. proposed the model and implemented the expriment. J.Z. was the research advisor and provided suggestions. W.G. provided guidance for this paper and contributed to the revisions. Z.M. completed data preprocessing. X.L. gave advice on optimization issues. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Social Science Fund Project Grant (No. 23BXW074).

**Data Availability Statement:** MR (Movie Review) is a movie review dataset. Each sample is a movie review, which express the emotional tendency of observers. The dataset contains 10,662 samples in total, including 5331 positive emotional samples and 5331 negative emotional samples. SST (Stanford Sentiment Treebank) is also an emotion classification dataset. It is an extension of MR and is divided into two versions according to the number of emotion categories, that is SST1 and SST2. SST1 dataset has five types of tags, including 8544 training data, 1101 verification data and 2210 test data. SST2 dataset consists of two types of tags, including 6920 training data, 872 verification data and 1821 test data. TREC (Text Retrieval Conference) is a question classification dataset. It contains 6 question labels and consists of 5452 training data and 500 test data. Subj is a subjective data set. Its task is to divide sentences into subjective and objective. Subj consists of 10,000 samples and is trained by ten-fold cross-validation. The URL of MR is https://www.cs.cornell.edu/people/pabo/movie-review-data/. The URL of SST1 and SST2 is https://nlp.stanford.edu/sentiment/. The URL of TREC is https://nlp.stanford.edu/sentiment/. The URL of Subj is www.cs.cornell.edu/people/pabo/movie-review-data/ accessed on 21 October 2023.

**Acknowledgments:** We would like to thank the above fund for their technical and financial support.

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

## References


1. Quazi, S.; Musa, S.M. Text Classification and Categorization through Deep Learning. In Proceedings of the 14th International Conference on Computational Intelligence and Communication Networks, Al-Khobar, Saudi Arabia, 4–6 December 2022; pp. 513–519.
2. Uriarte-Arcia, A.V.; López-Yáñez, I.; Yáñez-Márquez, C. One-Hot Vector Hybrid Associative Classifier for Medical Data Classification. *PLoS ONE* **2014**, *9*, e95715. [CrossRef]
3. Parida, U.; Nayak, M.; Nayak, A.K. Ranking of Odia Text Document Relevant to User Query Using Vector Space Model. In Proceedings of the 2019 International Conference on Applied Machine Learning, Bhubaneswar, India, 25–26 May 2019; pp. 165–169.
4. Chen, S.; Bolufé-Röhler, A.; Montgomery, J.; Zhang, W.; Hendtlass T. Using Average-Fitness Based Selection to Combat the Curse of Dimensionality. In Proceedings of the 2022 IEEE Congress on Evolutionary Computation, Padua, Italy, 18–23 July 2022; pp. 1–8.
5. Sumarsono, A. Application of RXD Algorithm to Word Vector Representation for Keyword Identification. In Proceedings of the 10th Annual Computing and Communication Workshop and Conference, Las Vegas, NV, USA, 6–8 January 2020; pp. 306–310.
6. Dogan, G.; Ergen, B. A new mobile convolutional neural network-based approach for pixel-wise road surface crack detection. *Measurement* **2022**, *195*, 111119–111123. [CrossRef]
7. Dhyani, M.; Kumar, R. An intelligent Chatbot using deep learning with Bidirectional RNN and attention model. *Mater. Today* **2021**, *34*, 817–824. [CrossRef] [PubMed]
8. Kim, Y. Convolutional Neural Networks for Sentence Classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1746–1751. [CrossRef]
9. Arevian, G. Recurrent Neural Networks for Robust Real-World Text Classification. In Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI'07), Fremont, CA, USA, 2–5 November 2007; pp. 326–329.
10. You, H.; Yu, L.; Tian, S.; Ma, X.; Cai, W. MC-Net: Multiple max-pooling integration module and cross multi-scale deconvolution network. *Knowl.-Based Syst.* **2021**, *231*, 107456–107464. [CrossRef]
11. Yu, Y. Research on Music Emotion Classification Based on CNN-LSTM Network. In Proceedings of the 5th Asian Conference on Artificial Intelligence Technology (ACAIT), Haikou, China, 29–31 October 2021; pp. 473–476.
12. Lidong, H.; Hui, Z. A new short text sentimental classification method based on multi-mixed convolutional neural network. In Proceedings of the 3rd International Conference on Cloud Computing and Big Data Analysis, Chengdu, China, 20–22 April 2018; pp. 93–99.
13. Zhou, C.; Sun, C.; Liu, Z.; Lau, F.C.M. A C-LSTM Neural Network for Text Classification. *Comput. Sci.* **2015**, *4*, 39–44.
14. Vijayaprabakaran, K.; Sathiyamurthy, K. Towards activation function search for long short-term model network: A differential evolution based approach. *J. King Saud Univ.-Comput. Inf. Sci.* **2020**, *34*, 2637–2650. [CrossRef]
15. Prakash, S.; Jalal, A.S.; Pathak, P. Forecasting COVID-19 Pandemic using Prophet, LSTM, hybrid GRU-LSTM, CNN-LSTM, Bi-LSTM and Stacked-LSTM for India. In Proceedings of the 6th International Conference on Information Systems and Computer Networks, Mathura, India, 1–6 March 2023. [CrossRef]
16. Lai, S.; Lei, D. Calculation of sentence vector similarity based on fasttext model of weighted fusion. In Proceedings of the 4th International Conference on Advances in Computer Technology, Information Science and Communications (CTISC), Suzhou, China, 22–24 April 2022; pp. 1–6.
17. Lu, W.; Duan, Y.; Song, Y. Self-Attention-Based Convolutional Neural Networks for Sentence Classification. In Proceedings of the 6th International Conference on Computer and Communications (ICCC), Chengdu, China, 11–14 December 2020; pp. 2065–2069.
18. Yin, W.; Schütze, H. Multichannel Variable-Size Convolution for Sentence Classification. *arXiv* **2016**. [CrossRef]
19. Wang, C.; Wang, B.; Xu, M. Tree-Structured Neural Networks With Topic Attention for Social Emotion Classification. *IEEE Access* **2019**, *7*, 95505–95515. [CrossRef]
20. Yang, X.; Liu, X. Convolutional Recurrent neural network with attention mechanism based improved skip-gram algorithm for text sentiment classification. In Proceedings of the 7th International Conference on Information Science and Control Engineering, Changsha, China, 18–20 December 2020; pp. 410–414.
21. Xian, T.; Wen, L.; Yi, D.; Ting, W. Short Text Feature Extraction and Classification Based on Serial-Parallel Convolutional Gated Recurrent Neural Network. *Adv. Eng. Sci.* **2019**, *51*, 125–132.
22. Cheng, Z.; Tong, H.; Man, X. BLSTM_MLPCNN Model for Short Text Classification. *Comput. Sci.* **2019**, *46*, 206–211.

23. Li, J.; Yang, X. A Cyclical Learning Rate Method in Deep Learning Training. In Proceedings of the International Conference on Computer, Information and Telecommunication Systems (CITS), Hangzhou, China, 5–7 October 2020; pp. 1–5.
24. Saravanan, V.; Ranjana, P. Stochastic Gradient Descent on Modern Hardware for Business Environment. In Proceedings of the 7th International Conference on Intelligent Computing and Control Systems, Madurai, India, 17–19 May 2023; pp. 812–815.