symmetry

MDPI

Article

A Text Classification Model via Multi-Level Semantic Features

Keji Mao [1,†], Jinyu Xu [1,†], Xingda Yao [1], Jiefan Qiu [1], Kaikai Chi [1] and Guanglin Dai [2,*]

[1] College of Computer Science and Technology College of Software, Zhejiang University of Technology, Hangzhou 310023, China
[2] College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China
* Correspondence: dgl@zjut.edu.cn
† These authors contributed equally to this work.

Abstract: Text classification is a major task of NLP (Natural Language Processing) and has been the focus of attention for years. News classification as a branch of text classification is characterized by complex structure, large amounts of information and long text length, which in turn leads to a decrease in the accuracy of classification. To improve the classification accuracy of Chinese news texts, we present a text classification model based on multi-level semantic features. First, we add the category correlation coefficient to TF-IDF (Term Frequency-Inverse Document Frequency) and the frequency concentration coefficient to CHI (Chi-Square), and extract the keyword semantic features with the improved algorithm. Then, we extract local semantic features with TextCNN with symmetric-channel and global semantic information from a BiLSTM with attention. Finally, we fuse the three semantic features for the prediction of text categories. The results of experiments on THUCNews, LTNews and MCNews show that our presented method is highly accurate, with 98.01%, 90.95% and 94.24% accuracy, respectively. With model parameters two magnitudes smaller than Bert, the improvements relative to the baseline Bert+FC are 1.27%, 1.2%, and 2.81%, respectively.

Keywords: BiLSTM; keyword extraction; symmetry and asymmetry; machine learning; TextCNN with symmetric-channel

check for updates

Citation: Mao, K.; Xu, J.; Yao, X.; Qiu, J.; Chi, K.; Dai, G. A Text Classification Model via Multi-Level Semantic Features. *Symmetry* **2022**, *14*, 1938. https://doi.org/10.3390/sym14091938

Academic Editor: José Carlos R. Alcantud

Received: 23 August 2022
Accepted: 14 September 2022
Published: 17 September 2022

1. Introduction

Information has entered the Big Bang stage as the Internet era has evolved. Especially with the rapid development and popularization of multi-media technology, people can access massive amounts of uncertain information in their daily lives. Text is the main way of carrying modern multimedia information. Therefore, mining, analyzing and classifying textual information are now mainstream tasks in NLP. Text classification is a major task of NLP. According to the scenario and the content of text classification, it is mainly classified into sentiment classification [1,2], news classification [3,4], topic classification [5], Q&A matching [6] and so on. As a major product of the information age, news text is characterized by a large amount of data, high real-time requirements, complicated manual tagging etc. In order to promptly place news into corresponding categories and reduce human workload, a model that is both fast and has high accuracy is needed.

Methods based on machine learning (especially, deep learning) have been fruitful in NLP in the last few years. On the one hand, CNNs (Convolutional Neural Networks) have been commonly used in NLP, such as VGG [7] and ResNet [8], because of the local perceptual field and weight sharing [9]. On the other hand, RNNs (Recurrent Neural Networks) are also an important technique in NLP because of their ability to obtain contextual information, such as LSTM [10,11], GRU [12] etc.

Yoon Kim [13] presented TextCNN in 2014, where text is processed by a single-layer CNN and convolutional kernels of different sizes after vectorizing the text with word2vec [14,15]. The reliability of the method was demonstrated on sentiment analysis and problem classification tasks, and provided evidence that unsupervised word vector pre-training has significant implications in NLP deep learning. However, the obvious problem is that it only focuses on local information and does not consider the context, where some fuzzy words may not mean the same thing literally and in different contexts. "Apple", as an example, might be a fruit, but it might also be a technology company.

Compared with CNN, RNN's output values are dependent on the output values of the previous moment and the input values of this moment. It can be applied to capture the information in the context of the text. However, the biggest drawback of RNN in the training process is gradient explosion. Therefore, researchers have presented improved RNN models, such as LSTM and GRU, which prevent the occurrence of the main drawback in RNNs by selectively forgetting some information. BiLSTM [16], which was presented later as a variant of LSTM, has a good development in NLP. It has become one of the mainstream models in the NLP field so far by using a bidirectional mechanism, i.e., superimposing the forward LSTM with the backward LSTM, while including contextual semantic information.

Google presented a pre-training model called Bert [17] in 2018, which is a pre-training model based on Transformers [18] for a bidirectional encoder. An MLM (masked language model) was used to generate deep bidirectional language representations, which achieved SOTA on 11 NLP tasks upon publication. Although its results are good, it has two main drawbacks; one is the large scale of the model with numerous parameters, and the other is the existence of a limit on the length of the input text, which cannot exceed at most 510 TOKENs, which reduces its accuracy when dealing with longer texts.

In summary, we present a text classification model based on multi-level semantic features to achieve a high accuracy rate while having small model size and few parameters. Experiments on Chinese news text classification datasets such as THUCNews, MCNews and LTNews show that the model has good classification results.

Our contribution is focused on the following three points.

- We present a text classification model based on multi-level semantic information. The multi-level semantic information mainly contains keyword semantic information, local semantic information and global semantic information. The keyword information is extracted by using the improved TF-IDF based on the category correlation coefficient and the improved CHI based on the frequency concentration coefficient, the local semantic information is extracted by using the improved symmetric-channel TextCNN, and the global semantic information is extracted by using the BiLSTM with attention;
- We present a symmetric-channel mechanism for enriching the local semantic information extracted by TextCNN. Experiments demonstrate that TextCNN has better results for text classification after adding the symmetric-channel mechanism;
- We improve the keyword extraction algorithm by adding the category correlation coefficient to TF-IDF and the frequency concentration coefficient to CHI. Experiments demonstrate that the improved keyword extraction algorithm can have better results for text classification.

2. Related Work

Research on text classification has flourished over the past few years. Some researchers focus on the individual models themselves. Yoon Kim [13] presented TextCNN, which is a shallow and wide CNN. After vectorizing the text with Word2Vec, the text features are extracted by a single convolutional layer with convolutional kernels of different sizes. Huang et al. [19] presented DenseNet, which is a densely connected deep CNN. By referring to the idea of ResNet, the CNN structure is deepened by introducing the connection skipping mechanism. However, a problem that exists along with the deepening of the network is the growth of the number of parameters and the model volume. Le H T et al. [20] compared two different types of CNNs on several datasets by different input methods. One type is shallow and wide CNNs represented by TextCNN, and the other is deep CNNs represented by DenseNet. It was tentatively concluded through experiments that deep CNNs are not able to be more effective than shallow CNNs in text classification. Li J. et al. [21]

presented the BiLSTM model with hierarchical attention. Layers of attention are added behind the BiLSTM layer, a word-level attention layer and a sentence-level attention layer, for analyzing the importance of words and sentences. Wang B [22] presented a disconnected RNN. He incorporates position invariance into the RNN. By limiting the distance of RNN information flow, the hidden state of each time step is restricted to words near the current position, which is an improvement compared to RNN and CNN.

Some researchers have focused on multi-model fusion. Deng J. et al. [23] presented the attention-based BiLSTM fused CNN with a gating mechanism model. The model first calculates the context vector through the attention mechanism, then captures the contextual features through BiLSTM, then captures the salient features of the topic through CNN, and finally eliminates other topic-related information through the gating mechanism. Zhang J et al. [24] presented a feature fusion text classification model combining CNN and BiGRU with a multi-attention mechanism. Global semantic features are extracted by BiGRU, local semantic features are extracted by CNN, and the features are stitched together and then classified. Xu F. et al. [25] presented a multi-level semantic feature extraction algorithm. Character features are extracted using CNN and word features are extracted using BiGRU. After connecting the two features, local semantic features and contextual semantic features are extracted by LightTextCNN and BiLSTM, respectively. Finally, the four features are fused for classification.

Some researchers optimized the pre-trained word vector model. Google AI Research Institute presented a pre-training model called Bert [17] in 2018, which is a pre-training model based on the bi-directional encoder of Transformers. Once published, it achieved SOTA on 11 NLP tasks. Qiu Y. et al. [26] used TF-IDF to find the weight of each word and weighted the word vector obtained from the Word2Vec model. The weighted word vectors are classified better than the cases of VSM, LDA and Word2Vec alone. Zhao W. et al. [27] first derived the LDA topic information matrix and the TD-IDF weight matrix, and mixed the two matrices and weighted them into the word vector extracted by Word2Vec. The classification effect was verified on several datasets.

In summary, research on model fusion has been relatively hot in recent years. However, since the underlying models are still only a few existing mainstream models, mainly CNNs and RNNs, the development has been relatively limited, mostly in the form of different combinations of several models. Part of the research is devoted to deeper mining of the text in an attempt to extract more kinds of information from the limited text. There are two main problems with the existing research. One is that the models with high accuracy basically use Bert and its variants, which also leads to the large size of the models themselves. The second is that the models with small size do not have high relative accuracy.

Based on the current state of research, we present a text classification model based on multi-level semantic features in order to have a high accuracy rate while having a small model size and few parameters. Sometimes, simple methods also have better results at lower costs, so we chose the TF-IDF and CHI algorithms, which are relatively simple, as the basis of the keyword extraction algorithm. Most studies have weighted the Word2Vec model after calculating the weights from TF-IDF [28]. Unlike those studies, we extract the keywords of the text with the improved TF-IDF and the improved CHI, and incorporate the keywords as a new feature into the model. TextCNN, as a member of CNN, is often used as a baseline for comparison in NLP due to its excellent local information extraction ability. Therefore, this paper improves TextCNN to enhance its local information extraction ability. We add a symmetric-channel mechanism to TextCNN to extract local semantic information. Then, we use BiLSTM with attention to extract global semantic information, because BilSTM is the best RNN model to extract global semantic information at this stage. Finally, we classify the text after fusing the three sets of semantic information. Each feature contains semantic information in different dimensions. A statistically based approach extracts keyword features, which are a set of lexical sequences that have a positive effect on the classification. The machine learning-based approach extracted local features as well as global features, which exploits the local information focus property of CNN and the global information inclusion property of RNN, respectively. When features of different dimensions are aggregated together, they can have a positive and mutually reinforcing effect on text classification. Results of the experiments show that our presented method has competitive results.

3. Model

The structure of the text classification model based on multi-level semantic features presented in this research is shown in Figure 1. The model is divided into three main modules, which are the keyword semantic extraction module, the local semantic extraction module and the global semantic extraction module. The keyword semantic extraction module first uses TF-IDF based on the category correlation coefficient and CHI based on the frequency concentration coefficient to train the importance dictionary on the training set. Then, the text is divided by Jieba and the TopK keywords are filtered based on the importance dictionary. After merging the keywords obtained by the two algorithms, the keyword sequences are transformed into word vectors using a Word2Vec pre-training model. Finally, the keyword semantic features are obtained by single-layer one-dimensional convolution and an FC (fully-connected) operation is performed. After the text has been divided into words by Jieba, the local semantic module and the global semantic module transform the text sequences into word vectors using a Word2Vec pre-training model. The local semantic information is extracted using the symmetric-channel-based TextCNN and the global semantic information is extracted using the attention-based BiLSTM, respectively.

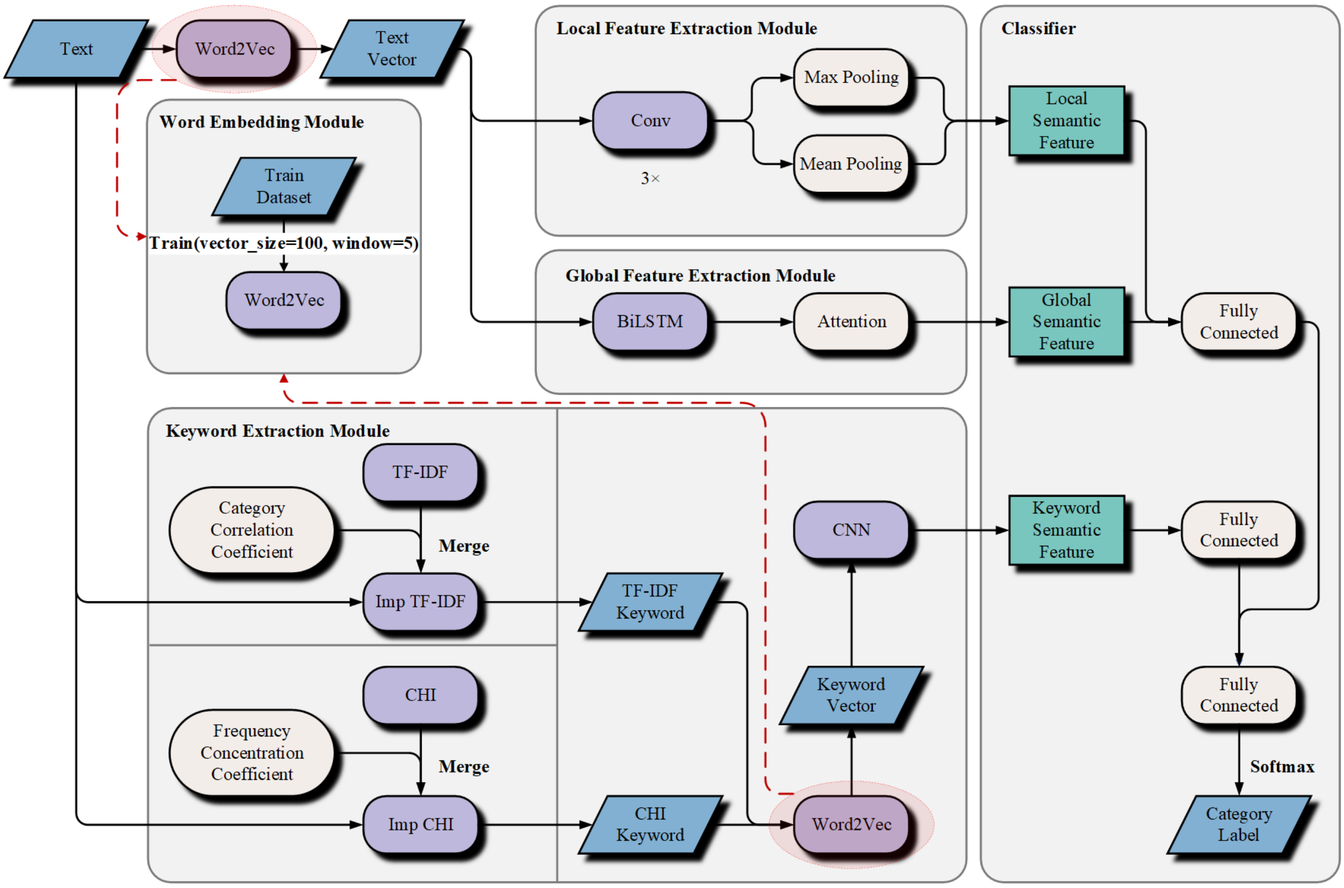

Figure 1. Structure of the text classification model based on multi-level semantic features.

3.1. *Word Embedding Module*

It has been shown that using a pre-trained word vector model can improve text classification accuracy and decrease the dimensionality of the word vector. Common word vector models include Word2Vec, Glove [29], Bert [17] etc. Considering the number of parameters and the model size, Word2Vec is used as the word vector pre-training model in this paper.

In the earliest days, One-Hot was used as a word vector, but this had serious problems, namely dimensional explosion and semantic gaps. All words in One-Hot need one dimension to be stored, and the size of the word vector dimension is the size of the whole vocabulary. Therefore, when the number of words in the corpus is larger, the dimensionality of One-Hot is also larger, and the computation required in processing is horrible, which is the reason for the dimensional explosion. On the other hand, each word is represented as an independent dimension and there is no direct connection between dimensions. Thus, even if two words have similar meanings, their One-Hot representations may be worlds apart, i.e., a semantic divide.

Therefore, in 1986, Hinton G.E. [30] presented a distributed representation, which is a dense vector of fixed length. The main idea is that the semantic meaning of words is determined by contextual information, and words that appear in similar contexts have similar semantics. It reveals the semantic connections between words while reducing the word vector dimensionality. Word2Vec is a typically distributed representation. Since Mikolov T. et al. presented Word2Vec in 2013, Word2Vec has been one of the most common word vector representations in NLP.

There are two main training methods for Word2Vec, namely Skip-gram and CBOW (Continuous Bag-of-Words). Skip-gram is trained by predicting the context by the current word, which is equivalent to giving a word and guessing what words may come before and after it. CBOW is trained by predicting the current word from the context. It is equivalent to removing a word from a sentence and guessing what that word is.

3.2. Keyword Extraction Module

Keyword features are extracted from the text using the improved TF-IDF and CHI. We believe that a small amount of text contains information about the text as a whole, i.e., from a small amount of text, we can infer the category of the text. TF-IDF and CHI are used to analyze the importance weight of each word on the training set in a statistical way. The importance weights are used to extract the sequence of words in a piece of text that can help infer the category of the text, i.e., keywords.

3.2.1. Improved TF-IDF

TF-IDF [28] is a concept presented by Jones K.S. as a statistical metric in information retrieval to assess the importance of a word to one of the documents in a corpus. The main idea of the algorithm is to assume that the more frequently a word appears in one document and the less frequently it appears in other documents, the more important this word is for this document and the more suitable it is for use in classification. In practice, it is often applied as a weighting technique. The calculation is shown in Equations (1)–(3).

$$\mathrm{TF}(v,d) = \frac{n_d(v)}{n_d(V)} \tag{1}$$

$$\mathrm{IDF}(v) = \lg\left(\frac{|\mathrm{D}|}{|\{m : v \in d_m\}| + 1}\right) \tag{2}$$

$$\text{TF-IDF}(v,d) = \mathrm{TF}(v,d) \times \mathrm{IDF}(v) \tag{3}$$

v refers to an arbitrary vocabulary. V refers to the set of all vocabulary. d refers to an arbitrary document. D refers to a corpus. $v \in V$ and $d \in D$. $\mathrm{TF}(v,d)$ refers to the frequency of occurrence of v in d. $n_d(v)$ refers to the number of occurrences of v in d. $n_d(V)$ refers to the total number of occurrences of all vocabulary in d. $\mathrm{IDF}(v)$ refers to the inverse document frequency, which is an indicator of the general importance of v. $|\mathrm{D}|$ refers to the total number of documents in D. $|\{m : v \in d_m\}|$ refers to the total number of documents in which v appears. Because there may be cases where the vocabulary does not appear in all documents, it is necessary to add 1 to $|\{m : v \in d_m\}|$ in order to prevent arithmetic errors. Eventually, multiplying $\mathrm{TF}(v, d)$ with $\mathrm{IDF}(v)$ yields $\mathrm{TF\text{-}IDF}(v, d)$.

For the original TF-IDF, the main problem is that its process of calculating the inverse document probability only takes into consideration the relation between v and the number of documents it appears in ,and ignores the differences in the distribution between different categories that are most important for classification. Therefore, we present an improved TF-IDF that incorporates a category correlation coefficient α in the calculation of the inverse text probability. The validity of the category correlation coefficient is experimentally demonstrated. Its calculation is shown in Equations (4)–(6).

$$\lambda_i = \max\left(\frac{|\{n : v \in d_n, d_n \in c_j, j \neq i\}|}{|d : d \in c_j, j \neq i|}\right) \tag{4}$$

$$\alpha_i = \frac{|\{m : v \in d_m, d_m \in c_i\}|}{|d : d \in c_i|} - \lambda_i \tag{5}$$

$$\alpha = \max(\alpha_i) \tag{6}$$

$max(x)$ refers to the formula for calculating the maximum value. λ_i refers to the maximum value of the intra-class probability of occurrence of v in all categories except c_i. c_i refers to the i-th category. α_i refers to the category correlation coefficient of c_i. α is the maximum value of the category correlation coefficient for each category. The formula looks complicated, but the main idea is to obtain the category correlation coefficient α by calculating the intra-category occurrence probability of v in each categorical category and then using the largest intra-category occurrence probability minus the second-largest intra-category occurrence probability. The calculation of the improved TF-IDF is shown in Equation (7).

$$\mathrm{TF\text{-}IDF}(v, d) = \mathrm{TF}(v, d) \times \alpha \times \mathrm{IDF}(v) \tag{7}$$

We believe that the category correlation coefficient α contains the relationship between each vocabulary and category, which solves the problem of the original TF-IDF to some extent.

3.2.2. Improved CHI

CHI, also known as Chi-Square, was first presented by Karl Pearson in 1900 [31]. The main idea is to assume that the values obey a certain distribution, calculate the expectation of the values under that distribution, compare the variance of the true and expected values, and finally calculate the difference between the true and expected values. The simple way to understand this is that when the calculated CHI value is larger, then the hypothesis is not valid and the hypothesis is rejected. On the contrary, when the calculated CHI value is smaller, the hypothesis is valid and the hypothesis is accepted. When applied to the field of text classification, we first assume that v and text categories are independent of each other. On this basis, when the calculated CHI value is larger, then the hypothesis is rejected and the more related v and the text category are. Conversely, when the calculated CHI value is smaller, then the hypothesis is accepted and the less relevant v and the text category are. Taking binary classification, for instance, the data distribution is shown in Table 1.

Table 1. Distribution table of binary classification data.

Vocabulary	c_1	c_2	Total
v	A	B	A + B
$\neg v$	C	D	C + D
Total	A + C	B + D	N

The expectation of c_1 containing v is calculated as shown in Equation (8).

$$E(v, c_1) = \frac{(A + C) \times (A + B)}{N} \tag{8}$$

The variance of c_1 containing v is calculated as shown in Equation (9).

$$\begin{aligned} D(v, c_1) &= \frac{(A - E(v, c_1))^2}{E(v, c_1)} \\ &= \frac{(A \times D - B \times C)^2}{N \times (A + C) \times (A + B)} \end{aligned} \tag{9}$$

The variance calculation for the remaining terms is similar to Equation (9). The final CHI calculation is shown in Equation (10).

$$\begin{aligned} \chi^2(v, c_1) &= D(v, c_1) + D(v, c_2) + D(\neg v, c_1) + D(\neg v, c_2) \\ &= \frac{N \times (A \times D - B \times C)^2}{(A + C) \times (A + B) \times (B + D) \times (C + D)} \end{aligned} \tag{10}$$

For the original CHI, the main problem is that it only counts the number of documents in which v appears in its calculation without considering the number of occurrences of v. This can lead to the algorithm being biased towards low-frequency vocabulary. For example, suppose that v_1 appears 1 time in all documents of c_1, while v_2 appears 10 times in 99% of the documents of c_1. In normal thinking, v_2 is more important for classification. However, based on the original CHI, $\chi^2(v_1, c_1)$ will be greater than $\chi^2(v_2, c_1)$. This can lead to the omission of very important feature terms. Therefore, we present an improved CHI by adding a frequency concentration coefficient β to the calculation process. The validity of the frequency concentration coefficient is experimentally demonstrated. The calculation procedure is shown in Equations (11)–(14).

$$s = \text{sum}\left(\frac{\text{sum}(TF(v, d_n))}{|\{n : v \in d_n, d_n \in c_j, j \neq i\}|}\right) \tag{11}$$

$$\delta_i = \frac{s}{|\{j : c_j \in C, j \neq i\}|} \tag{12}$$

$$\beta_i = \frac{\text{sum}(TF(v, d_m))}{|\{m : v \in d_m, d_m \in c_i\}|} - \delta_i \tag{13}$$

$$\beta = \max(\beta_i) \tag{14}$$

$sum(x)$ refers to the formula for calculating the sum. $max(x)$ refers to the formula for calculating the maximum value. c_i refers to the i-th category. δ_i refers to the average frequency of occurrence of v in all categories except c_i. β_i refers to the frequency concentration coefficient of c_i. β is the maximum value of the frequency concentration factor for each category. The formula looks complicated; the main idea is to calculate the average frequency of occurrence of v in c_i, and then subtract the average frequency of occurrence of v in all other categories, and what we obtain is the coefficient of c_i. Taking the maximum value of the coefficients for all categories, the final frequency concentration coefficient β is obtained. The calculation of the improved CHI is shown in Equation (15).

$$\chi^2(v, c_1) = \frac{\beta \times N \times (A \times D - B \times C)^2}{(A + C) \times (A + B) \times (B + D) \times (C + D)} \tag{15}$$

We believe that the frequency concentration coefficient β contains the relationship between vocabulary and frequency, which solves the problem of the original CHI to some extent.

3.3. Local Feature Extraction Module

In the local feature extraction module, we use TextCNN [13], which is a CNN with a single layer of multiple convolutional kernels. The local semantic information comes from machine learning, not statistics. The model is allowed to train a set of model parameters on the training set to fit the text classification under that dataset. Because of the nature of CNNs, the model will focus on the local aspects of the text, scaling up the local features. In order to extract richer local semantic information, we add a symmetric-channel mechanism to TextCNN; i.e., we symmetrically use max-pooling and mean-pooling in the pooling layer after the convolution layer. The structure of the TextCNN based on the symmetric-channel mechanism that we use is shown in Figure 2.

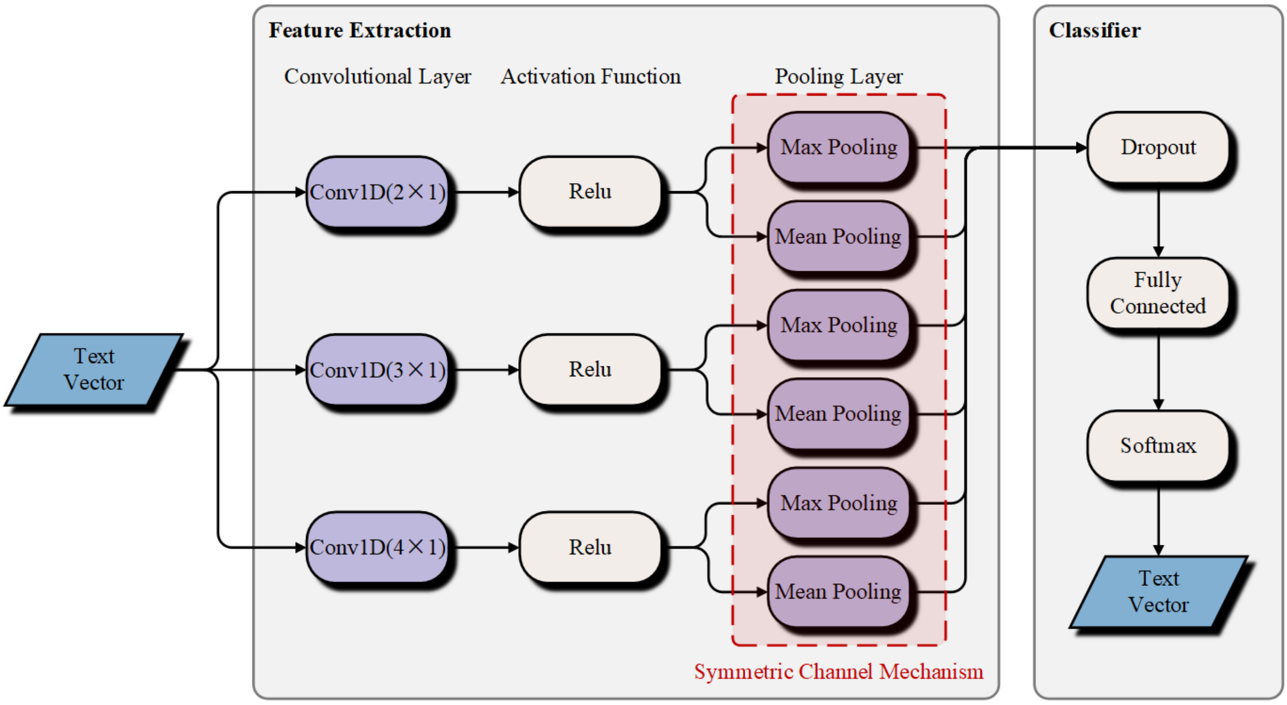

Figure 2. Structure of TextCNN.

The text is first transformed into word vectors using Word2Vec. Local semantic features of word vector sequences are collected by convolutional kernels of different sizes. The local semantic information is then enriched by the symmetric-channel mechanism. Then, the collected local semantic information is concatenated together. Finally, the output values are connected to a FC layer to obtain the logit, and the logit is connected to a Softmax layer for classification. The Relu activation function is designed to prevent the gradients from vanishing. The role of Dropout [32] is to prevent overfitting. We believe that using both max-pooling and mean-pooling for feature acquisition on text can capture as much local semantic information as possible when the text length is limited. Results of experiments show that after using the symmetric-channel mechanism, the text classification effectiveness of TextCNN is improved.

3.4. Global Feature Extraction Module

In the global feature extraction module, we use BiLSTM, a variant model of LSTM [10,11]. In BiLSTM, the output at each moment contains the semantic information in context, so it is equivalent to obtaining the global semantic information of the text. LSTM is a special kind of RNN. The structure of RNN is shown in Figure 3. RNN can retain the previous information to the present. However, as training proceeds, RNN will not be able to effectively take advantage of the previous information. In this case, LSTM emerges. The major role of LSTM is to deal with gradient disappearance and gradient explosion.

The major distinction from LSTM to RNN, as shown in Figure 4, consists of three main gates [33,34].

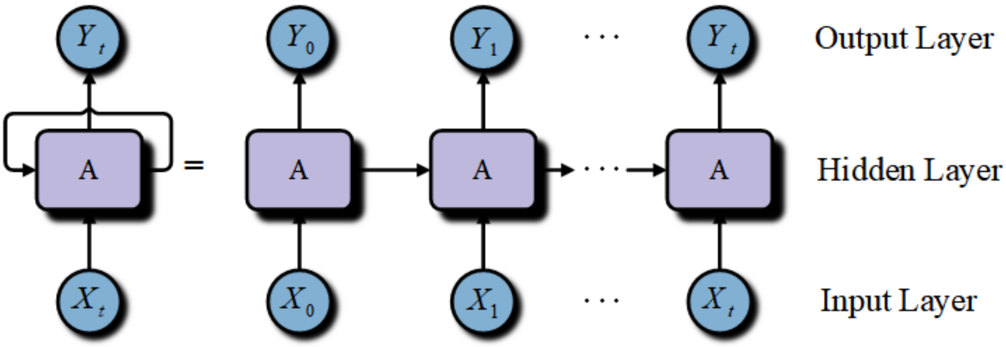

Figure 3. Structure of RNN [35].

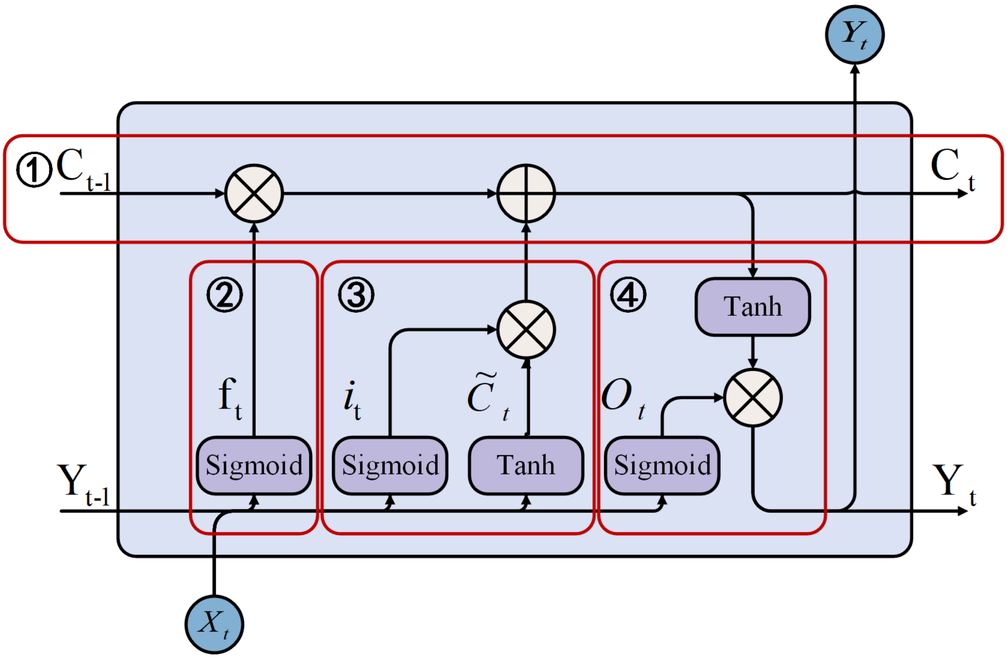

Figure 4. Structure of LSTM block A [35].

The first part is the cell state. It conveys the control message from previous to present and then to the future. The control message is C_t. C_t is updated as shown in Equation (16). C_{t-1} is the control message of previous. C_t is always transferred for memorization purposes.

$$C_t = f_t \times C_{t-1} + i_t \times \widetilde{C_t} \tag{16}$$

The second part is the forgetting gate. It is composed by a *sigmoid* operation. The *sigmoid* operation outputs a vector with every digit between zero and one. The closer the digit is to one, the more information will be received, and one represents complete acceptance. A digit closer to zero represents more information forgotten, with zero representing totally forgetting. The vector, i.e., f_t, can decide which information should be conveyed to the cell state. Its formula is given in Equation (17). W_f refers to the weight of the forgetting gate. σ refers to the *sigmoid* operation.

$$f_t = \sigma\left(W_{fy} \times Y_{t-1} + W_{fx} \times X_t + b_f\right) \tag{17}$$

The third part is the input gate. It is composed by a *sigmoid* operation, a *tanh* operation and an element multiplication operation. An output value is generated by the *tanh* operation and the *sigmoid* operation. Eventually, this output is added to the cell state. The formula is given in Equations (18) and (19). i_t refers to the output of the *sigmoid* operation. $\widetilde{C}_t$ refers to the output of the *tanh* operation. W_i refers to the weight of the input gate. Finally, the two output values are multiplied element by element.

$$i_t = \sigma\left(W_{iy} \times Y_{t-1} + W_{ix} \times X_t + b_i\right) \tag{18}$$

$$\widetilde{C}_t = \tanh\left(W_{Cy} \times Y_{t-1} + W_{Cx} \times X_t + b_C\right) \tag{19}$$

The fourth part is the output gate. It is composed by a *sigmoid* operation, a *tanh* operation and an element multiplication operation. The *tanh* operation is intended to stabilize the values by compressing the control message learned previously. The *sigmoid* operation is intended to obtain the output from the input, regardless of the previously learned information. Eventually, the output value is obtained through the multiplication of the outputs. The formula is given in Equations (20) and (21). o_t refers to the output of the *sigmoid* operation. Eventually, the output of the LSTM cell is obtained by the element multiplication of the control message C_t and the output of the *sigmoid* operation after the *tanh* operation.

$$o_t = \sigma\left(W_{oy} \times Y_{t-1} + W_{ox} \times X_t + b_o\right) \tag{20}$$

$$Y_t = o_t \times \tanh(C_t) \tag{21}$$

The main idea of BiLSTM, as a variant of LSTM, is to connect the same input sequence into the forward and backward LSTMs respectively, and then connect the corresponding outputs of the two networks together to jointly access the output layer for prediction. The role of attention is to allocate weights for the global semantics, so that the global semantic information comes with a focus. The structure of attention-based BiLSTM is shown in Figure 5.

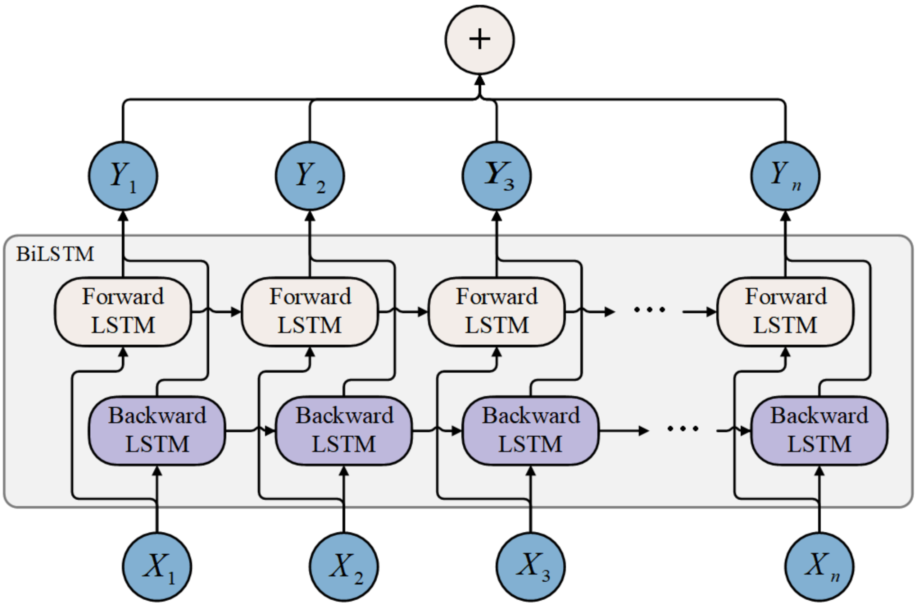

Figure 5. Structure of attention-based BiLSTM.

4. Experiment

In order to improve the accuracy of Chinese news long text classification, we present a text classification model based on multi-level semantic information. In this section, we conducted experiments with THUCNews, LTNews and MCNews in order to validate the efficiency of our presented method.

4.1. Evaluation Metrics

To validate the effect of various methods, we used confusion matrices including TP (True Positive), TN (True Negative), FP (False Positive) and FN (False Negative). Four evaluation metrics were used, namely Accuracy, WP (Weighted Precision), WR (Weighted Recall) and WF1 (Weighted F1) [36]. The calculation is shown in Equations (22)–(25). N refers to the total number of samples. m refers to the total number of categories.

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{m}(TP_i + TN_i) \tag{22}$$

$$WP = \frac{1}{N}\sum_{i=1}^{m} n_i \frac{TP_i}{TP_i + FP_i} \tag{23}$$

$$WR = \frac{1}{N}\sum_{i=1}^{m} n_i \frac{TP_i}{TP_i + FN_i} \tag{24}$$

$$WF1 = \frac{1}{N}\sum_{i=1}^{m} n_i \frac{2P_iR_i}{P_i + R_i} \tag{25}$$

4.2. Dataset

The datasets we used were THUCNews, LTNews and MCNews. All experiments were performed on these three datasets. Their parameters are shown in Table 2.

Table 2. Table of dataset parameters.

Dataset	Average Length	Longest Length	Shortest Length	Number	Category
THUCNews(Train)	913	27,467	8	50,000	10
THUCNews(Dev)	881	10,919	15	5000	10
THUCNews(Test)	969	14,720	13	10,000	10
LTNews(Train)	3180	40,902	2001	10,000	10
LTNews(Dev)	3163	60,032	2001	1000	10
LTNews(Test)	3225	54,349	2001	2000	10
MCNews(Train)	1027	7414	8	10,239	9
MCNews(Dev)	1091	6876	9	4389	9

4.2.1. THUCNews

This dataset was obtained from http://thuctc.thunlp.org/ [37], accessed on 16 November 2021. We used 10 of these categories with more data as the experimental data set. In this paper, two datasets are derived from the original dataset. One is called the normal dataset and the other is called the extra-long dataset. The normal dataset is 6500 data extracted separately from each category in the source dataset. The extra-long text dataset is 1300 data extracted separately from each category in the source dataset, which are all over 2000 in length. Both of them are divided into a training set, a validation set and a test set at the ratio of 7:1:2. We named this extra-long dataset based on THUCNews "LTNews".

4.2.2. MCNews

This dataset was obtained from http://www.cnsoftbei.com/, accessed on 23 May 2022. It is the A7 question dataset of the 10th China Software Cup. The dataset has nine categories and contains 14,631 data. The training and validation sets are divided at a ratio of 7:3. The data distribution is shown in Figure 6. The figure shows that the dataset is extremely unevenly distributed, making it difficult for practical classification.

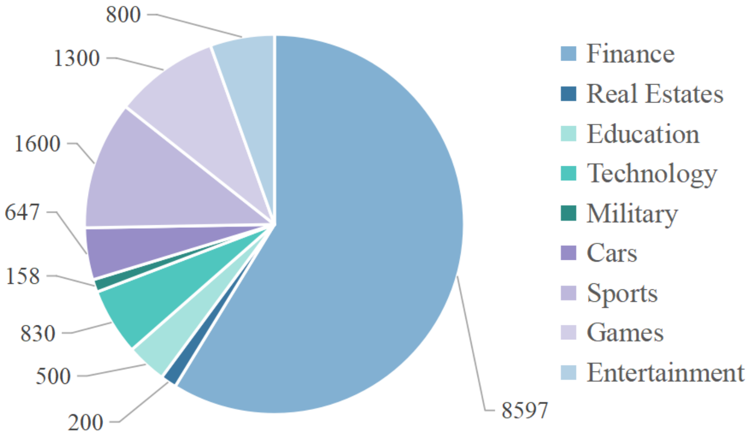

Figure 6. Data distribution of MCNews.

4.3. Experimental Environment and Parameters

The experiments were all run in Python 3.7, TensorFlow 1.14 and CUDA 10.0. The main hardware was an Intel(R) Core(TM) i5-8400 CPU @ 2.80 GHz (2808 MHz) and a NVIDIA GeForce GTX 1050 (2048 MB). Bert as a baseline was trained on a NVIDIA GeForce GTX 2080Ti (11 GB). The early stop mechanism, which stops the training, will be used in case there is no optimization after 10 validation sets.

In this paper, Word2Vec is used as a pre-trained word vector model and the trained word vector has a dimension of 100. The word vector model is trained in CBOW training mode by calling the Word2Vec function in Python's gensim library with a window size of 5. The data used for training is the training set for each dataset. The specific training parameters are shown in Table 3. Previously, we performed a large number of experiments for the determination of the model parameters. Several comparison experiments were conducted in a certain range to select the best set of model parameters. However, due to the limitation of space and our belief that this part of the experiments is of lower relative importance, the model parameters are directly presented in the table.

Table 3. Table of training parameters.

Name of Parameter	Value
Word2Vec dimension	100
Sequence length	200 (common) 600 (used in LTNews)
Num. of TextCNN filters	128
Kernel size	[2, 3, 4]
Num. of BiLSTM hidden	128
Dropout rate	0.5
Learning rate	1×10^{-3}
Batch size	32
Num. of epochs	10
Num. of keywords	20

4.4. Ablation Study

4.4.1. Symmetric Channel Mechanism

We present a symmetric-channel mechanism, i.e., using both max-pooling and mean-pooling in the pooling layer of TextCNN to obtain richer local semantic information. We validated our method on THUCNews, LTNews and MCNews, and the results of the experiment can be seen in Table 4 and Figure 7.

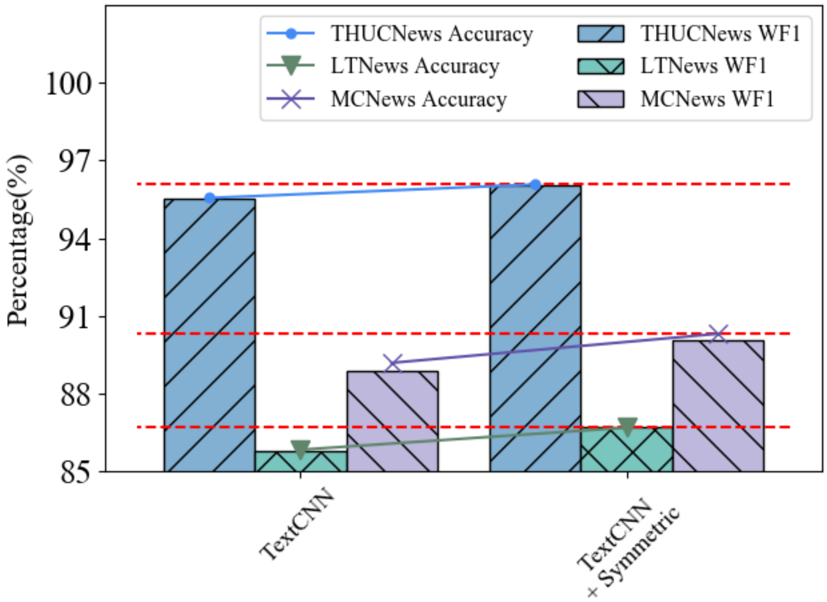

Figure 7. Results of the symmetric-channel mechanism.

Table 4. Table of experimental results of the symmetric-channel mechanism.

Model	THUCNews		LTNews		MCNews	
	Accuracy	**WF1**	**Accuracy**	**WF1**	**Accuracy**	**WF1**
TextCNN	95.56	95.55	85.85	85.81	89.20	88.87
TextCNN+Symmetric	**96.08**	**96.06**	**86.70**	**86.74**	**90.32**	**90.07**

From the experimental results, the classification effect of TextCNN was significantly improved after adding the symmetric-channel mechanism, with 0.52%, 0.85% and 1.12% improvement on THUCNews, LTNews and MCNews, respectively. This is because after adding the symmetric channel mechanism, the originally extracted local semantic features are supplemented, enriching the local semantic information and increasing the volume of the model in disguise. For neural networks, the classification ability of models built with larger model volumes is more powerful within a certain range. It is experimentally demonstrated that the local semantic extraction capability of TextCNN can indeed be increased with the addition of the symmetric-channel mechanism.

4.4.2. Multi-Level Semantic Fusion

We present a text classification model based on multi-level semantics, i.e., extracting keyword features using a keyword extraction algorithm, extracting local features using TextCNN, extracting global features using BiLSTM, and fusing multi-level semantic information to obtain improved classification effects. We validated our method on THUCNews, LTNews and MCNews, and the results of the experiment can be seen in Table 5 and Figure 8.

From the experimental results, just a simple combination of TextCNN and BiLSTM together achieved better results than either model alone. This is the basis of our research, which illustrates that the fusion of multiple different levels of semantic information can provide better classification capabilities for the model. Furthermore, a classification capability close to that of using Bert+FC was achieved. The classification ability of the model is further improved with the use of the symmetric channel mechanism, which again illustrates the effectiveness of the symmetric channel mechanism.

Finally, we added the keyword features extracted by the TF-IDF to the model. After fusing keyword semantic information, local semantic information, and global semantic information, the accuracy rates improved by 1.45%, 3.4%, and 3.26% compared to TextCNN, and

1.15%, 2.35%, and 2.05% compared to BiLSTM, respectively. Compared with the adopted base models, i.e., TextCNN and BiLSTM, there is a large improvement. The accuracy of fusing the three types of semantic information was significantly improved compared to fusing two types of semantic information. This shows that adding keyword features to the model does have a positive contribution to the classification ability of the model.

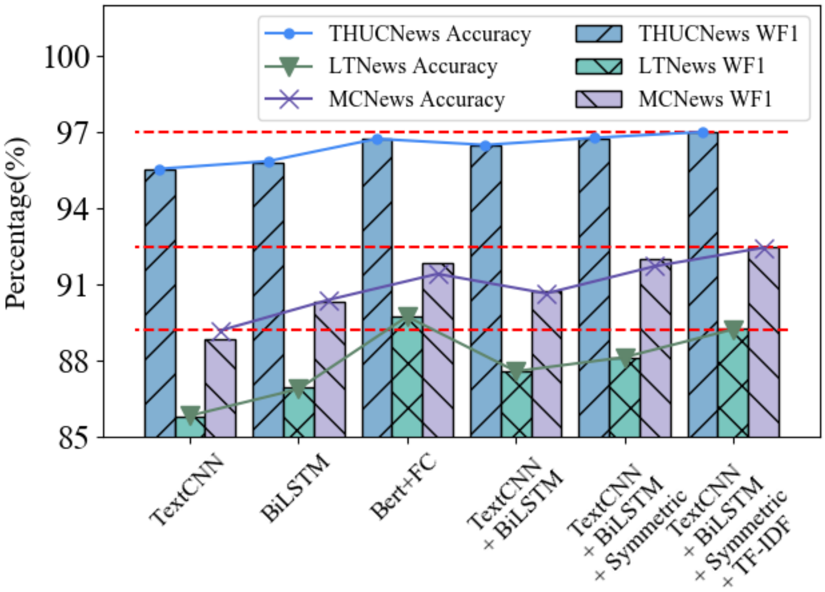

Figure 8. Results of multi-level semantic fusion.

Table 5. Table of experimental results of multi-level semantic fusion.

Model	THUCNews		LTNews		MCNews	
	Accuracy	WF1	Accuracy	WF1	Accuracy	WF1
TextCNN	95.56	95.55	85.85	85.81	89.20	88.87
BiLSTM	95.86	95.81	86.90	86.99	90.41	90.32
Bert+FC	96.74	96.72	**89.75**	**89.77**	91.43	91.87
TextCNN+BiLSTM	96.50	96.47	87.60	87.59	90.66	90.73
TextCNN+BiLSTM +Symmetric	96.78	96.76	88.15	88.10	91.73	91.99
TextCNN + BiLSTM +Symmetric+TF-IDF	**97.01**	**96.99**	89.25	89.26	**92.46**	**92.47**

The classification effect is not better on LTNews compared to Bert+FC because the effect of multi-level semantic fusion is based on the adopted base model. On LTNews, the effect of the underlying model is relatively poor, resulting in the results of multi-level semantic fusion not exceeding those of Bert+FC. On the other two datasets, our presented method outperforms Bert+FC.

4.4.3. Improved Keyword Extraction Algorithm

We improve the traditional keyword extraction algorithm, mainly choosing TF-IDF and CHI, so that the keywords extracted by the improved keyword extraction algorithm are more sufficient to obtain higher classification results. We validated our method on THUCNews, LTNews and MCNews, and the results of the experiment can be seen in Table 6 and Figure 9, where TBS indicates the use of TextCNN, BiLSTM and the symmetric-channel.

From the experimental results, after adding the category correlation coefficient to TF-IDF, the keyword features extracted by the improved algorithm have stronger classification ability compared with the original TF-IDF. After adding frequency concentration coefficients to CHI, the keyword features extracted by the improved algorithm also have stronger classification ability relative to the original CHI. The improved TF-IDF improved 0.38%, 1% and 1.25%, respectively, and the improved CHI improved 0.36%, 1.55% and 1.19%, respectively, with respect to the original algorithm. After improvement, the results of our presented method all exceeded Bert+FC. TY8s also directly demonstrates the effectiveness of our improvements to the original algorithm.

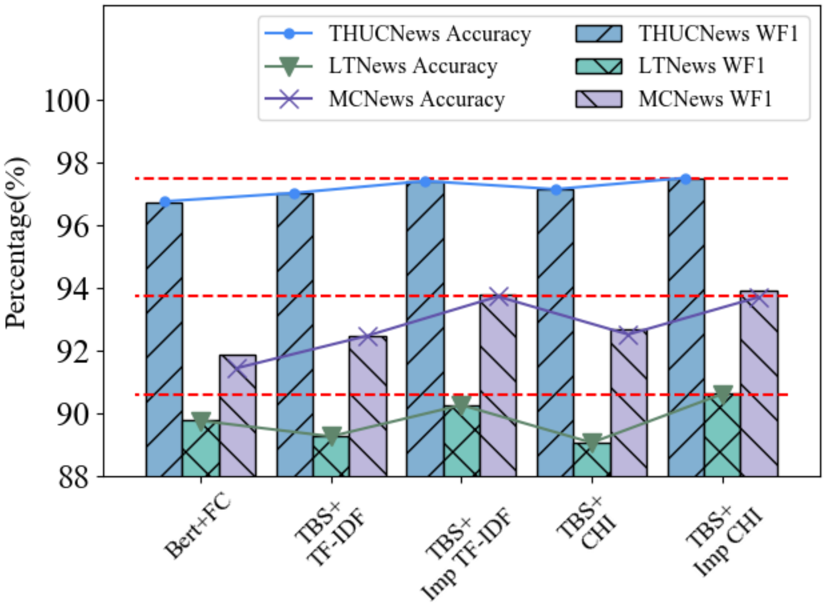

Figure 9. Results of the improved keyword algorithm.

Table 6. Table of experimental results of the improved keyword algorithm.

Model	THUCNews		LTNews		MCNews	
	Accuracy	WF1	Accuracy	WF1	Accuracy	WF1
Bert+FC	96.74	96.72	89.75	89.77	91.43	91.87
TBS+TF-IDF	97.01	96.99	89.25	89.26	92.46	92.47
TBS+Imp TF-IDF	97.39	97.38	90.25	90.23	93.71	93.78
TBS+CHI	97.13	97.12	89.05	89.03	92.50	92.66
TBS+Imp CHI	**97.49**	**97.49**	**90.60**	**90.61**	**93.69**	**93.91**

4.4.4. Keyword Fusion

We fuse two keyword features, namely TF-IDF features and CHI features, to obtain improved classification effects with the fused keyword features. We validated our method on THUCNews, LTNews and MCNews, and the results of the experiment can be seen in Table 7 and Figure 10, where TBS indicates the use of TextCNN, BiLSTM and the symmetric-channel.

From the experimental results, the two keyword features fused in this paper have better classification effects. Compared to the single keyword feature, results were improved by 0.62%, 0.7% and 0.53%, and 0.52%, 0.35% and 0.55%, respectively. This is because the fused keyword features combine the advantages of the two keyword extraction algorithms to obtain richer keyword features, which in turn leads to better classification results.

Table 7. Table of experimental results of keyword fusion.

Model	THUCNews		LTNews		MCNews	
	Accuracy	WF1	Accuracy	WF1	Accuracy	WF1
Bert+FC	96.74	96.72	89.75	89.77	91.43	91.87
TBS+Imp TF-IDF	97.39	97.38	90.25	90.23	93.71	93.78
TBS+Imp CHI	97.49	97.49	90.60	90.61	93.69	93.91
TBS+Imp TF-IDF +Imp CHI	**98.01**	**98.01**	**90.95**	**90.95**	**94.24**	**94.39**

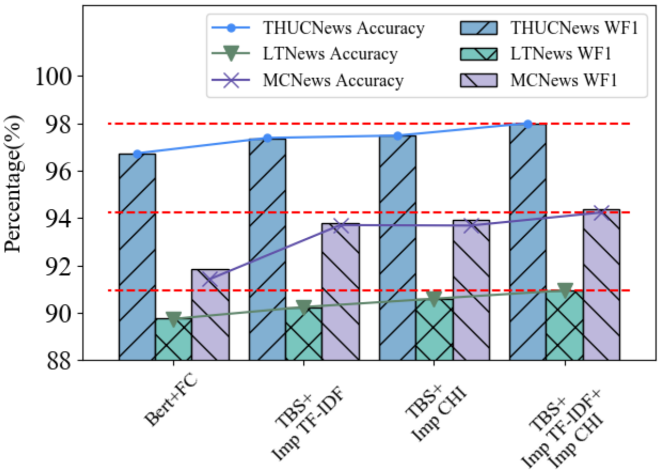

Figure 10. Results of keyword fusion.

4.5. Main Experimental Results

Due to the data selection, the number of categories and the categories selected, directly comparing the results of experiments is impossible, even under the same dataset. Therefore, this paper uses a relatively fair comparison with the baseline and the methods presented by other studies, as shown in Table 8. The "-" in the table represents data not given in the study. The comparison was made on THUCNews.

Table 8. Table of comparisons with other studies.

Contrast Mode	Model	Accuracy	WP	WR	WF1
Comparison with Bert+FC	TextCNN	−1.18	−1.14	−1.18	−1.17
	BiLSTM + attention	−0.88	−0.86	−0.88	−0.91
	Bert + FC	0	0	0	0
	LFCN [36]	1.2	-	-	-
	Model [38]	-	0.43	0.42	0.46
	TBAM [39]	-	-0.10	0.10	0.10
	Our model	**1.27**	**1.23**	**1.27**	**1.29**
Eight Categories	HCapsNet [40]	98.13	-	-	-
	WTL-CNN [27]	96.60	-	-	-
	Our model	**98.70**	-	-	-

From the experimental results, the accuracy of our presented method is improved by 1.2% relative to the baseline Bert+FC, and by 2.45% and 2.15% relative to TextCNN and BiLSTM, respectively. When compared with the methods presented in other studies, our presented method also has better results. Compared to other methods, we add the dimension of keyword features. We believe that the statistically based keyword sequence contains the main information about the text as a whole, which can help the model focus on the key information of the text during classification. Although using a pre-trained model of several orders of magnitude smaller in size, it was still able to have higher accuracy than other methods based on large pre-trained models. The comparison with WTL-CNN can show, to some extent, that using the weights obtained from TF-IDF to filter keywords for text classification directly is better than weighting the weights to word vectors. The above results prove the effectiveness and competitiveness of the text classification model based on multi-level semantic information presented in this paper.

4.6. Summary of Experimental Results

From the results of the ablation study, the symmetric-channel mechanism, multi-level semantic fusion, keyword extraction algorithm improvement and keyword fusion presented in this paper are practical and effective. Each improvement enhances the classification ability of the model, and all are positive and can complement each other's enhancements.

After using the symmetric-channel mechanism, the effect is improved relative to the original TextCNN. After using multi-level semantic fusion, the effect is improved relative to the normal fused TextCNN with BiLSTM. After using the improved keyword extraction algorithm, the effect is improved relative to the original keyword extraction algorithm. After using the keyword fusion, the effect is improved relative to the single keyword extraction algorithm. Although the effect of each improvement is small, the improvements can be stacked together to form a larger improvement in the end.

Experiments on several datasets show that the text classification model based on multi-level semantic features presented here outperforms the baseline Bert+FC by 1.27%, 1.2% and 2.81%, respectively, when the model parameters are two orders of magnitude smaller relative to Bert. Moreover, from the viewpoint of the attributes and distribution of the dataset, our presented method has the best results in the common case, the extra-long text case and the unbalanced text case, and has strong applicability as well as practicality.

5. Conclusions

In order to improve classification results while having a low number of parameters, we present a text classification model based on multi-level semantic features. First, we improve the keyword extraction algorithm by adding category correlation coefficients to TF-IDF and frequency concentration coefficients to CHI. The keyword sequences are extracted from the text as features by the improved keyword extraction algorithm. Then, we add a symmetric-channel to the TextCNN to enrich the extracted local semantic information. We take advantage of the fact that the features extracted by BiLSTM contain contextual information and we apply an attention mechanism to BiLSTM to extract the global semantic features of the text. Finally, the fused semantic features of the three semantic features are used for text category prediction. Through extensive experimental comparisons, our presented method has a large improvement over the base model, achieving the highest accuracies of 98.01%, 90.95% and 94.24% on THUCNews, LTNews and MCNews, indicating that our presented method is applicable in the common case, the extra-long text case, and the unbalanced text case. Furthermore, with model parameters two orders of magnitude smaller relative to Bert, the improvements relative to the baseline Bert+FC are 1.27%, 1.2% and 2.81%, respectively.

In this paper, we validate the effectiveness of our proposed method on news texts. This is only a preview of future work in our group, which has not been possible due to the fact that data collection has not begun until now. In the future, we will validate the applicability, as well as make subsequent improvements to the model proposed in this paper, on the text classification task of mental illness dialogues.

Author Contributions: Conceptualization, K.M. and J.X.; methodology, K.M. and J.X.; software, J.Q. and X.Y.; validation, J.X., J.Q., K.C. and G.D.; formal analysis, K.C. and X.Y.; investigation, J.X.; resources, K.M.; data curation, J.X., X.Y. and K.C.; writing—original draft preparation, J.X.; writing—review and editing, K.M. and K.C.; visualization, J.X. and J.Q.; supervision, G.D.; project administration, J.X., K.M. and G.D.; funding acquisition, K.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the Basic Public Welfare Research Project of Zhejiang Province [Grant Number LGG22F020014]; National Natural Science Foundation of China [Grant Number 62072410].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Wang, B.; Shan, D.; Fan, A.; Liu, L.; Gao, J. A Sentiment Classification Method of Web Social Media Based on Multidimensional and Multilevel Modeling. *IEEE Trans. Ind. Inform.* **2021**, *18*, 1240–1249. [CrossRef]
2. Zhou, Y.; Liao, L.; Gao, Y.; Wang, R.; Huang, H. TopicBERT: A topic-enhanced neural language model fine-tuned for sentiment classification. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**. [CrossRef]
3. Bhattacharya, P.; Patel, S.B.; Gupta, R.; Tanwar, S.; Rodrigues, J.J. SaTYa: Trusted Bi-LSTM-Based fake news classification scheme for smart community. *IEEE Trans. Comput. Soc. Syst.* **2021**. [CrossRef]
4. Al-Ahmad, B.; Al-Zoubi, A.; Abu Khurma, R.; Aljarah, I. An evolutionary fake news detection method for covid-19 pandemic information. *Symmetry* **2021**, *13*, 1091. [CrossRef]
5. Mao, S.; Zhang, L.L.; Guan, Z.G. An LSTM&Topic-CNN model for classification of online Chinese medical questions. *IEEE Access* **2021**, *9*, 52580–52589. [CrossRef]
6. Perevalov, A.; Both, A. Improving answer type classification quality through combined question answering datasets. In Proceedings of the International Conference on Knowledge Science, Engineering and Management, Tokyo, Japan, 14–16 August 2021; pp. 191–204. [CrossRef]
7. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
8. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
9. O'Shea, K.; Nash, R. An introduction to convolutional neural networks. *arXiv* **2015**, arXiv:1511.08458.
10. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
11. Rahman, M.M.; Watanobe, Y.; Nakamura, K. A bidirectional LSTM language model for code evaluation and repair. *Symmetry* **2021**, *13*, 247. [CrossRef]
12. Cho, K.; van Merrienboer, B.; Gülçehre, Ç.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv* **2014**, arXiv:1406.1078.
13. Kim, Y. Convolutional Neural Networks for Sentence Classification. *arXiv* **2014**, arXiv.1408.5882.
14. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, 3111–3119. Available online: https://proceedings.neurips.cc/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf (accessed on June 10, 2022).
15. Mikolov, T.; Corrado, G.; Kai, C.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
16. Zhou, P.; Shi, W.; Tian, J.; Qi, Z.; Li, B.; Hao, H.; Xu, B. Attention-based bidirectional long short-term memory networks for relation classification. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Berlin, Germany, 7–12 August 2016; pp. 207–212. [CrossRef]
17. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; Volume 1, pp. 4171–4186. [CrossRef]
18. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5998–6008. Available online: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf (accessed on June 10, 2022).
19. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708. [CrossRef]
20. Le, H.T.; Cerisara, C.; Denis, A. Do convolutional networks need to be deep for text classification? In Proceedings of the Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018. Available online: https://www.aaai.org/ocs/index.php/WS/AAAIW18/paper/view/16578/15542 (accessed on June 10, 2022).
21. Li, J.; Xu, Y.; Shi, H. Bidirectional LSTM with hierarchical attention for text classification. In Proceedings of the 2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chengdu, China, 20–22 December 2019; Volume 1, pp. 456–459. [CrossRef]
22. Wang, B. Disconnected recurrent neural networks for text categorization. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Melbourne, VI, Australia, 15–20 July 2018; pp. 2311–2320. [CrossRef]

23. Deng, J.; Cheng, L.; Wang, Z. Attention-based BiLSTM fused CNN with gating mechanism model for Chinese long text classification. *Comput. Speech Lang.* **2021**, *68*, 101182. [CrossRef]
24. Zhang, J.; Liu, F.; Xu, W.; Yu, H. Feature fusion text classification model combining CNN and BiGRU with multi-attention mechanism. *Future Internet* **2019**, *11*, 237. [CrossRef]
25. Xu, F.; Sun, S.; Xu, S.; Zhang, Z.; Chang, K.C. Chinese short text classification based on multi-level semantic feature extraction. In Proceedings of the International Conference on Advanced Intelligent Systems and Informatics, Cairo, Egypt, 11–13 December 2021; pp. 235–246. [CrossRef]
26. Qiu, Y.; Yang, B. Research on micro-blog text presentation model based on word2vec and tf-idf. In Proceedings of the 2021 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC), Dalian, China 14–16 April 2021; pp. 47–51. [CrossRef]
27. Zhao, W.; Zhu, L.; Wang, M.; Zhang, X.; Zhang, J. WTL-CNN: A news text classification method of convolutional neural network based on weighted word embedding. *Connect. Sci.* **2022**, *34*, 2291–2312. [CrossRef]
28. Jones, K.S. A statistical interpretation of term specificity and its application in retrieval. *J. Doc.* **1972**, *28*, 11–21. [CrossRef]
29. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543. [CrossRef]
30. Hinton, G.E. Learning distributed representations of concepts. In Proceedings of the Eighth Annual Conference of the Cognitive Science Society, Amgerst, Mass, 15–17 August 1986; Volume 1, p. 12.
31. Plackett, R.L. Karl Pearson and the chi-squared test. *Int. Stat. Rev. Int. Stat.* **1983**, *51*, 59–72. [CrossRef]
32. Hinton, G.E.; Srivastava, N.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R.R. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv* **2012**, arXiv:1207.0580.
33. Greff, K.; Srivastava, R.K.; Koutnik, J.; Steunebrink, B.R.; Schmidhuber, J. LSTM: A Search Space Odyssey. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *28*, 2222–2232. [CrossRef]
34. Zhao, Z.; Chen, W.; Wu, X.; Chen, P.C.Y.; Liu, J. LSTM network: A deep learning approach for Short-term traffic forecast. *IET Intell. Transp. Syst.* **2017**, *11*, 68–75. [CrossRef]
35. Mao, K.; Xu, J.; Jin, R.; Wang, Y.; Fang, K. A fast calibration algorithm for Non-Dispersive Infrared single channel carbon dioxide sensor based on deep learning. *Comput. Commun.* **2021**, *179*, 175 – 182. [CrossRef]
36. Chen, X.; Cong, P.; Lv, S. A Long-Text Classification Method of Chinese News Based on BERT and CNN. *IEEE Access* **2022**, *10*, 34046–34057. [CrossRef]
37. Sun M.; Li J.; Guo Z.; Zhao Y.; Zheng Y.; Si X.; Liu Z. THUCTC: An Efficient Chinese Text Classifier. **2016**
38. Zhang, M.; Shang, X. Chinese Short Text Classification by ERNIE Based on LTC_Block. *Wirel. Commun. Mob. Comput.* **2022**, *2022*. [CrossRef]
39. Liu, H.; Qian, Q. Bi-Level Attention Model with Topic Information for Classification. *IEEE Access* **2021**, *9*, 125366–125374. [CrossRef]
40. Li, Y.; Ye, M.; Hu, Q. *HCapsNet: A Text Classification Model Based on Hierarchical Capsule Network*; Springer: Berlin/Heidelberg, Germany, 2021; Volume 12816, pp. 538–549. [CrossRef]