future internet

MDPI

Essay

Chinese Text Classification Model Based on Deep Learning

Yue Li *, Xutao Wang and Pengjian Xu

School of Computer Science and Technology, Donghua University, Shanghai 201620, China; 2161704@mail.dhu.edu.cn (X.W.); 2171774@mail.dhu.edu.cn (P.X.)

* Correspondence: frankyueli@dhu.edu.cn

Received: 18 October 2018; Accepted: 12 November 2018; Published: 20 November 2018

Abstract: Text classification is of importance in natural language processing, as the massive text information containing huge amounts of value needs to be classified into different categories for further use. In order to better classify text, our paper tries to build a deep learning model which achieves better classification results in Chinese text than those of other researchers' models. After comparing different methods, long short-term memory (LSTM) and convolutional neural network (CNN) methods were selected as deep learning methods to classify Chinese text. LSTM is a special kind of recurrent neural network (RNN), which is capable of processing serialized information through its recurrent structure. By contrast, CNN has shown its ability to extract features from visual imagery. Therefore, two layers of LSTM and one layer of CNN were integrated to our new model: the BLSTM-C model (BLSTM stands for bi-directional long short-term memory while C stands for CNN.) LSTM was responsible for obtaining a sequence output based on past and future contexts, which was then input to the convolutional layer for extracting features. In our experiments, the proposed BLSTM-C model was evaluated in several ways. In the results, the model exhibited remarkable performance in text classification, especially in Chinese texts.

Keywords: Chinese text classification; long short-term memory; convolutional neural network

1. Introduction

Motivated by the development of Internet technology and the progress of mobile social networking platforms, the amount of textual information is growing rapidly on the Internet. Given the powerful real-time nature of Internet platforms, great potential value is hidden in such textual information; however, effective organization and management is in high demand presently. Text classification, known as an effective method for text information organization and management, is widely employed in the fields of information sorting [1], personalized news recommendation, sentiment analysis [2], spam filtering, user intention analysis, etc.

Machine-learning-based methods, including naive Bayes, support vector machine, and k-nearest neighbors, are generally adopted by traditional text classification. However, their performance depends mainly on the quality of hand-crafted features. Compared with the methods of machine learning, the method of deep learning proposed in 2006 is deemed as an effective method for feature extraction. Moreover, an increasing number of scholars have applied commonly used neural networks, including the convolutional neural network (CNN) and the recurrent neural network (RNN), to text classification.

Among the two, RNN has attained remarkable achievement in handling serialization tasks. As RNN is equipped with recurrent network structure which can be used to maintain information, it can better integrate information in certain contexts. For the purpose of avoiding the problem of gradient exploding or vanishing in a standard RNN, long short-term memory (LSTM) [3] and other variants [4] have been designed for the improvement of remembering and memory accesses. Living up

to expectations, LSTM does show a remarkable ability in the processing of natural language. Moreover, the other popular neural network, CNN, has also displayed a remarkable performance in computer vision [5], speech recognition, and natural language processing [6] because of its remarkable capability in capturing local correlations of spatial or temporal structures. In terms of natural language processing, CNN is able to extract n-gram features from different positions of a sentence through convolutional filters and then it learns both short- and long-range relations through the operations of pooling.

LSTM [3] is good at dealing with serialization tasks but poor in the ability to extract features, performs well in extracting features but lacks the ability to learn sequential correlations. Therefore, in the paper, both the CNN and the specific RNN architecture—bidirectional long short-term memory (BLSTM)—are combined together to establish a new model named as the BLSTM-C model for text classification. The BLSTM is employed firstly to capture the long-term sentence dependencies and then CNN is adopted to extract features for sequence modeling tasks. In addition, our model is evaluated by applying it to Chinese text classification. The model is applied to both the English and Chinese language and then corresponding effects are compared with each other. It turns out that our model is more suitable for the Chinese language. Furthermore, it is also shown through our evaluation that our BLSTM-C model achieves remarkable results and it also outperforms a wide range of baseline models.

2. Related Work

It is widely acknowledged that deep-learning-based neural network models have achieved great success in natural language processing. This paper focuses on establishing a new model that is able to obtain better results in the classification of Chinese text. To ensure that the computer can understand human language, the first step of text classification usually goes to representing the text with vectors which will later feed into the neural network. Therefore, the quality of the representation is doomed to play a quite significant role in the classification. For the final aim of obtaining a better representation of text, TF-IDF (Term Frequency–Inverse Document Frequency) and bag-of-words were employed in early research. Bag-of-words treats texts as unordered sets of words and each word of them is represented by a one-hot vector, a sparse vector in the size of the vocabulary, with 1 in the entry representing the word and 0 in all other entries [1]. Accordingly, this vector loses both the word order and syntactic feature. Mikolov came up with the idea of distributed representations of words and paragraphs, and it is shown by relevant experiments that word and paragraph vectors outperform bag-of-words models as well as other techniques for text representations [7,8].

In many works on text representation learning published recently, it is known that there are generally two popular neural network models that had achieved their remarkable performance—CNN and RNN.

The ability of CNNs [5] in extracting features from inputs, such as images, is outstanding and it has also achieved remarkable result in image classification. In such process, 2D convolution and 2D pooling operation are usually used to represent image input [5,9]. As for text input, Kalchbrenner utilized 1D convolution to perform feature mapping and then applied 1D max pooling operation over the time-step dimension to obtain a fixed-length output [6,10]. Moreover, in 2017, Conneau et al. adopted a very deep CNN in the tasks of text classification by pushing the depth to 29 convolutional layers [11].

RNN, as what is indicated by its name, is known as a kind of neural network that is equipped with a recurrent structure, for which RNN is capable of preserving sequence information over time. In addition, this feature enables that RNNs is applicable for serialization tasks such as text classification and speech recognition. Furthermore, Tai et al. [12] proposed a tree-LSTM, a variant of RNN allowing for richer network topologies where each LSTM unit is able to incorporate information collected from multiple child units. In addition, Zhou et al. [13] achieved success in extracting meaningful features from documents automatically by combining bi-directional LSTM with an attention mechanism.

Our work focuses mainly on Chinese text classification, which is known as a completely different language from English. In English, words are generally separated by spaces and an independent meaning is available for each word, while Chinese words, on the contrary, have no spaces to separate

them. In order to solve this problem, Zhang et al. [14] put forward an HHMM-based Chinese lexical analyzer, ICTCLAS, which actually showed the effectiveness of class-based segmentation HMM(Hidden Markov Model). Furthermore, Ma et al. employed a deep learning method—LSTM—to conduct Chinese word segmentation and achieved better accuracy in many popular datasets in comparison with the models based on more complex neural network architectures. As what is stated above, LSTM performs poorly in extracting features, while CNN lacks the ability to learn sequential correlations. In this paper, a new model integrating BLSTM with CNN is established for Chinese news classification. There has already been research that focuses on computer vision tasks such as image captions by combining CNN with LSTM. Moreover, Zhou et al. [15] also proposed a C-LSTM model that integrates CNN with an LSTM network for sentence modeling. Most of these models firstly apply CNN to data inputting to extract features and then feed them to the LSTM layer. However, the approach adopted by us is totally different: bi-directional LSTM is employed capture long-term sentence dependencies and then CNN is applied to extracting features for classification. Our experiment on the classification of Chinese news shows that our model outperforms the other related sequence-based models.

3. Hybrid Deep-Learning Model

The whole process of classification is shown in Figure 1. This chapter will describe each layer in the subsections and combine them as the BLSTM-C model at the end of this chapter.

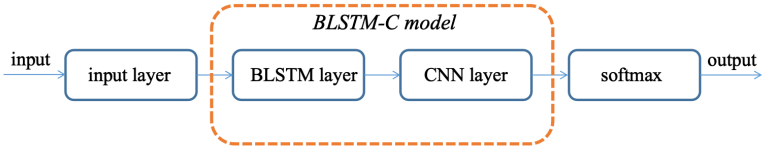

Figure 1. Classification process.

3.1. Input Layer

The first layer is the input layer that does some processing on the input. This layer can be divided into two parts: remove the stop words and segmentation. The first step removes the stop words from the article so that the information in the text will be more concentrated. The second step is a unique processing for Chinese language. Since Chinese words do not have a space to separate them, it would be necessary to segment them into words manually so that we can represent each word as vector.

3.1.1. Remove The Stop Words

In computing, stop words are words which are filtered out before or after processing of natural language data (text). Any group of words can be chosen as the stop words for a given purpose. For some search engines, these are some of the most common, short function words, such as 'the', 'is', 'at', 'which', and so on.

3.1.2. Segmentation

Chinese language is a unique language that is completely different from English. In English, word and word are separated by space and each word stands for independent meaning. On the contrary, Chinese words do not have a space to separate them. Furthermore, although each word has its independent meaning, their meaning is changed when the words are put together. Therefore, it is important and difficult to separate the words based on the context. Wrong Segmentation will totally change the sentence's meaning and increase the difficulty of classification.

After comparing the most commonly used tools for Chinese word segmentation, we finally choose "Jieba", which is built to be the best Python Chinese word segmentation module. The mainly algorithms for it are as follows:

- Achieve efficient word graph scanning based on a prefix dictionary structure [16]. Build a directed acyclic graph (DAG) for all possible word combinations.
- Use dynamic programming to find the most probable combination based on the word frequency.
- For unknown words, an HMM-based model [17] is used with the Viterbi algorithm.

For the Chinese words, it uses four states (BEMS) to distinguish them: B(begin), E(end), M(middle), and S(single). In addition, after training on large quantities of corpora, it gets three probability tables: TransProbMatrix, Emission Probability Matrix, and Initial State Matrix. Then, for a sentence that needs to be segmented, the HMM model uses a Viterbi algorithm to obtain the best 'BEMS' sequence that begins with a 'B'-word and ends with an 'E'-word.

Assume that, given the HMM state space S, there are k states, the probability of the initial state i is π_i, and the transition probability from the state i to the state j is a_{ij}. Let the observed output be y_1, ..., y_T. The most likely state sequence x_1, ..., x_T that produced the observation is given by the recurrence relation:

$$V_{i,j} = P(y_l|k) \cdot \pi_k, \tag{1}$$

$$V_{t,k} = max_{x \in S}(P(y_t|k) \cdot a_{x,k} \cdot V_{t-1,x}). \tag{2}$$

Here, $V_{t,k}$ is the probability of the most likely state sequence to correspond to the first t observations with a final state of k. The Viterbi path can be obtained by saving the backward pointer to remember the state x used in the second equation and declaring a function $Prt(k,t)$ that returns the x value to be used if $t > 1$ or returns a k value if $t = 1$:

$$x_T = argmax_{x \in S}(V_{T,x}), \tag{3}$$

$$x_{t-1} = Prt(x_t, t). \tag{4}$$

3.2. BLSTM Layer

Long Short-term Memory (LSTM) [3] is developed on the basis of Recurrent Neural Network (RNN) to solve the problems related to gradient vanishing or exploding. The mainly idea used by it is adding "gate" to the Recurrent Neural Network for the ultimate purpose of controlling the passing data. Generally speaking, a common architecture of LSTM units is composed of a memory cell, an input gate, an output gate and a forget gate. LSTM is shown in the form of a chain that is constructed by repeating modules of neural networks. With information stored inside, the memory cell runs across the whole chain. In addition, the other three gates are mainly designed to control whether to add or block the information to the memory cell.

With the output from h_{t-1}, the old hidden state, and the input from x_t, the current moment, the gates determine how to update the current memory cell and h_t, the current hidden state. The output of the forget gate represents the proportion of information that will be kept, while the input gate is mainly responsible for the addition of information to the memory cell. Moreover, this addition operation is made up of three parts. Firstly, a sigmoid layer is used to regulate the information that is required to be added to the memory cell. Secondly, the *tanh* function is adopted to obtain a vector through h_{t-1} and x_t. Finally, multiply these two values obtained and feed them to the memory cell. The output gate is responsible for the task of selecting useful information from the memory cell to output. For this purpose, it should firstly create a vector by applying the *tanh* function to the cell state and then regulate the information from h_{t-1} to x_t and multiply it by the vector created before. In this way, the output for this moment is obtained.

The LSTM transition functions are defined as follows:

$$i_t = \sigma(W_t \, [h_{t-1}, x_t] + b_i), \tag{5}$$

$$f_t = \sigma(W_f \, [h_{t-1}, x_t] + b_f), \tag{6}$$

$$o_t = \sigma(W_o\,[h\cdot, x\cdot] + b_o), \tag{7}$$

$$\tilde{c_t} = tanh(W_c[h_{t-1}, x_t] + b_f), \tag{8}$$

$$c_t = f_t \bullet c_{t-1} + i_t \bullet \tilde{c_t}, \tag{9}$$

$$h_t = o_t \bullet tanh(c_t). \tag{10}$$

σ refers to the logistic sigmoid function that has an output in [0, 1], *tanh* indicates the hyperbolic tangent function that has an output in $[-1, 1]$, and $\bullet$ denotes the elementwise multiplication. At the current time t, h_t refers to the hidden state, f_t represents the forget gate, i_t indicates the input gate, and o_t denotes the output gate. W_t, W_o, and W_f represent the weight of these three gates, respectively, while b_t, b_o, b_f refers to the biases of the gates.

As for BLSTM, it is regarded as an extension of the undirectional LSTM, and it not only adds another hidden layer but also connects with the first hidden layer in the opposite temporal order. Because of its structure, BLSTM can process the information from both the past and the future. Therefore, BLSTM is adopted to capture the information of the text input in this paper.

3.3. Convolutional Neural Networks Layer

There are only two dimensions in the input, among which $x_i \in \Re^d$ represents the d-dimensional vector for the i-th word in the sentence, while $x_i \in \Re^d$ denotes the input sentence. Moreover, L refers to the length of the sentence. Furthermore, one-dimensional convolution is employed to extract features from the output of LSTM layer.

A filter sliding over the input sequence is adopted by the one-dimensional convolution to detect features from different positions. Vector of the word in the sliding filter is denoted by x_j, x_{j+1}, $\cdots$, x_{j+k-1}, respectively. The window vector can be represented by the formula as follows:

$$W_j = [x_j, x_{j+1}, \cdots, x_{j+k-1}]. \tag{11}$$

The window vectors that are related to with the word x_j are: w_{j-k+1}, w_{j-k+2}, $\cdots$, and w_j, respectively. For each w_j, window vector, its feature map can be expressed by the formula follows:

$$c_j = f(w_j \circ m + b), \tag{12}$$

where $\circ$ refers to the dot product, $b \in R$ represents a bias term and f denotes a nonlinear transformation function that can be sigmoid, and hyperbolic tangent, etc. In our experiment, ReLU is chosen as the nonlinear function. In our model, n filters are adopted by us to produce the feature maps as follows:

$$\mathbf{W} = [c_1, c_2, \cdots, c_n]. \tag{13}$$

In the formula above, c_i refers to the feature map generated by the i-th filter. The convolution layer may have multiple filters of the same size to learn complementary features, or multiple kinds of filters with different sizes.

Then, a max-over pooling is applied to this feature map for the purpose of obtaining a vector of fixed length for classification. As for the pooling operation, it is adopted to extract maximum value from the matrix (feature map). After each convolution, a max-pool layer is added to extract the most significant elements in each convolution and then they are turned into feature vectors. It is common to periodically insert a pooling layer in-between successive Conv layers in a ConvNet architecture, which can progressively reduce the spatial size of the representation so as to reduce the amount of parameters and computation in the network, thus controlling the overfitting. Two common pooling operations, max pooling and average pooling, are shown in Figure 2. Max pooling chooses the max value in the filter as the new value in the new matrix while average pooling adopts the average value of all the numbers existing in the filter.

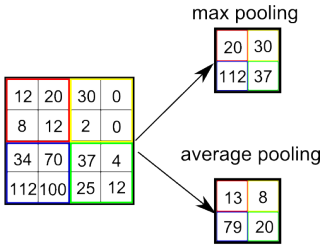

Figure 2. Pooling operation.

3.4. Proposed BLSTM-C Model

As shown in Figure 3, our model begins with a BLSTM layer to obtain a sequence output on the basis of the past context and the future context. Then, this sequence is fed to the CNN layer that is utilized to extract features from the previous sequence. After that, a max-over pooling layer is adopted to obtain a fixed length vector that is fed to the output layer that employs softmax function to classify the input. Blocks of the same color in the feature map layer and the window feature sequence layer correspond to features of the same window.

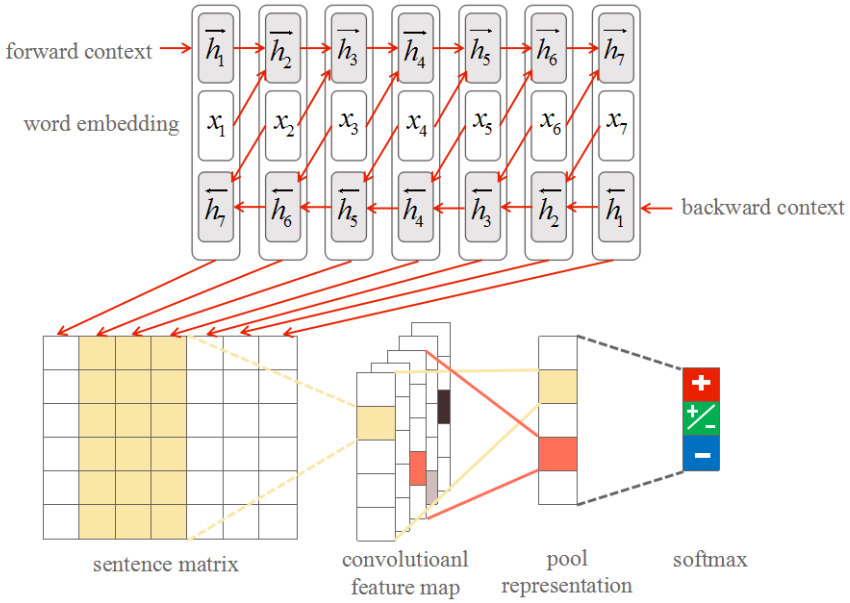

Figure 3. The architecture of the BLSTM-C model.

In the theory of probability, the output of the softmax function can be employed to represent a categorical distribution, that is, a probability distribution over K (number of different possible outcomes). As for our experiment, the probability distribution obtained will be over the categories of dataset. Moreover, the biggest one is the category that this input text belongs to. The function is shown as follows:

$$\sigma : \mathbb{R}^K \to \left\{ z \in \mathbb{R}^K \mid z_i > 0, \sum_{i=1}^{K} = 1 \right\}, \tag{14}$$

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}. \tag{15}$$

4. Experiment

4.1. Datasets

Our BLSTM-C model is evaluated on three datasets, and the summary statistics of the datasets obtained are shown as follows:

SST-1: Stanford Sentiment Treebank benchmark from Socher et al. [2]. This dataset is made up of 11,855 movie reviews and the reviews are split into train (8544), dev (1101) and test (2210), which aims to classify a review with fine-grained labels (very negative, negative, neutral, positive, and very positive).

SST-2: The same as SST-1, but the neutral reviews in it are removed and the binary labels (positive, negative) are adopted.

THUCNews: THUCNews is generated according to the historical data obtained from the subscription channel of Sina News RSS from 2005 to 2011. Based on the original classification system of Sina news, it is reintegrated and divided.

BBC: The English news dataset is originated from BBC news and it is adopted as the benchmarks for the research on machine learning. It is composed of 2225 documents from the BBC news website corresponding to stories from five topical areas from 2004 to 2005, including business, entertainment, politics, sport, and tech.

In all, eight categories of articles are selected by the main Chinese classification experiment for classification and they include politics, economy, stock market, technology, sports, education, fashion, and games, respectively.

For the comparison experiment on Chinese and English, five categories of articles are selected and they include business, entertainment, politics, sport, and tech.

4.2. Word Vector Initialization and Padding

Firstly, word2vec is employed to pre-train on large unannotated corpora. Through this way, better generalization can be achieved on the basis of limited amount of training data. In addition, maxlen is also set up to denote the maximum length of the sentence. Then, for every sentence, the stopwords are removed and the top maxlen words occurring in the word2vec model are transformed into vectors. For those sentences which are shorter than maxlen, they are padded with '0' vectors by us. In this way, the fixed-length input can be obtained for our model.

4.3. Hyper-Parameter Setting

For each dataset, 60% of the articles are randomly selected for training, 10% for validation and 30% for test. Moreover, the hyper-parameters are set up as follows.

The dimension of word vector is 300 for English word since the pre-trained word2vec model is chosen from Google for English, while the dimension of word vector for Chinese word is 250 because better representation on this configuration is obtained when the Chinese word2vec model is being trained by ourselves.

The maxlen of the sentence for SST-1 and SST-2 is 18 while the same parameter for THUCNews is 100. This maxlen parameter is established on the basis of the average length of the articles in each dataset. After lots of experiments, one BLSTM layer and one Convolutional layer are adopted by us when building our BLSTM-C model for all tasks. For the BLSTM layer, 50 hidden units are employed and the dropout rate obtained is 0.5. For Convolution layer, 64 convolutional filters with the window size of 5, and 1D pooling with the size 4 are employed.

5. Results

As shown in Table 1, our model is compared with 14 well-performed models from different tasks. One of the tasks is sentiment classification (SST-1, SST-2) while the other one is category classification (THUCNews).

5.1. Overall Performance

Both SST-1 and SST-2 datasets are employed to compare the performance of different methods. As shown in Table 1, our model is compared with some well-performed models from different areas, such as Support Vector Machine(SVM), Recursive Neural Network, Convolutional Neural Network,

and Recurrent Neural Network. Specifically, for the Recursive Neural Network, what are chosen include MV-RNN: Semantic Compositionality through Recursive Matrix-Vector Spaces [18], RNTN: Recursive Deep Models for semantic compositionality over a sentiment treebank [2], and DRNN: Deep Recursive Neural Networks for compositionality in language [19]. For CNN, what are chosen include DCNN: a CNN for modeling sentences [6], CNN-nonstatic and CNN multichannel: Convolutional Neural Networks for sentence classification [10], and Molding-CNN: Molding CNNs for text, including nonlinear and non-consecutive convolutions [20]. For Recurrent Neural Networks, what are chosen include RCNN: Recurrent Convolutional Neural Networks for Text Classification [20], S-LSTM: Long Short-term Memory over recursive structures [21], BLSTM and Tree-LSTM: improved semantic representations from tree-structured Long Short-term Memory networks [12]. For other baseline methods, we use a Support Vector Machine, n-gram bag of words and a Paragraph Vector. In addition, LSTM and B-LSTM were also implemented by us for further comparison of category classification on our Chinese news dataset.

Table 1. Comparison with baseline models on Stanford Sentiment Treebank and THUCNews.

Model	SST-1(%)	SST-1(%)	THUCNews(%)	Reported in
SVM	40.7	79.4	77.5	Socher et al.(2013b) [2]
NBoW	42.4	80.5	75.5	Le and Mikolov (2014) [8]
Paragraph Vector	48.7	48.7	48.7	Le and Mikolov (2014) [8]
DCNN	48.5	86.8	-	Kalchbrenner et al. (2014) [6]
CNN-non-static	48.0	87.2	-	Kim (2014) [10]
Modeling-CNN	**51.2**	88.6	-	Lei et al. (2015) [20]
CNN-multichannel	47.4	88.1	-	Kim (2014) [10]
RCNN	47.21	-	-	Lai et al. (2015) [20]
S-LSTM	-	81.9	-	Zhu et al. (2015) [21]
BLSTM	49.1	87.5	-	Tai et al. (2015) [12]
Tree-LSTM	51.0	87.5	-	Tai et al. (2015) [12]
MV-RNN	44.4	82.9	-	Socher et al. (2012) [18]
RNTN	45.7	85.4	-	Socher et al. (2013b) [2]
DRNN	49.8	86.6	-	Irsoy and Cardie (2014) [19]
LSTM	47.1	87.0	83.4	Our implementation
B-LSTM	47.3	88.1	86.5	Our implementation
CNN	46.5	85.5	82.5	Our implementation
BLSTM-C	50.2	**89.5**	**90.8**	Our implementation

The table above shows that our BLSTM-C model achieves remarkable performance in two out of three tasks (numbers in bold represent the best results). In sentiment classification, our BLSTM-C model gets the best result in the SST-2 dataset while molding-CNN achieves the best performance in the SST-1 dataset. Although our model fails to beat the state-of-art ones, it still obtains an acceptable result which means that the model is feasible for various scenarios.

As for the classification of text category, our model outperforms other well-performed models, achieving outstanding results. Through the comparison between our model and single-layer LSTM, B-LSTM and LSTM models, it is found that our model does combine the advantages of both LSTM and CNN. Apart from successfully learning long-term dependencies, it extracts features from text, leading to better results. Although almost no human-designed features are employed in our model, it beats the state-of-the-art SVM that highly requires engineered features.

5.2. Comparison between English and Chinese

To validate the ability of our model on different languages, our BLSTM-C model is compared with a simple LSTM model on the English news dataset as well as the Chinese news dataset that has the same categories as the English one. The five categories include technique, sports, business,

entertainment and politics. For an English experiment, a Google pre-trained word2vec model is chosen to represent the English words with vectors. The English news dataset is originated from BBC news and it mainly serves as the benchmarks for machine learning research. It is composed of 2225 documents from the BBC news website, corresponding to stories from five topical areas from 2004 to 2005.

Simple LSTM is also adopted by us to compare the improvement of our BLSTM-C model on Chinese dataset with that of English dataset. Tables 2 and 3 show the results obtained from English while Tables 4 and 5 present the results gained from Chinese. The number displayed in the table represents the number of articles that are classified as within this category.

Table 2. Output of the simple LSTM on the English dataset.

	Tech	**Sports**	**Politics**	**Entertainment**	**Business**
Tech	145	2	1	46	13
Sports	0	264	0	0	0
Politics	1	6	178	2	7
Entertainment	2	0	0	180	0
Business	2	1	6	3	254
Accuracy					91.73%

Table 3. Output of the BLSTM-C on the English dataset.

	Tech	**Sports**	**Politics**	**Entertainment**	**Business**
Tech	187	0	0	9	11
Sports	1	261	0	1	1
Politics	2	1	179	0	12
Entertainment	4	0	2	175	1
Business	5	0	7	0	254
Accuracy					94.88%

Table 4. Output of the simple LSTM on the Chinese dataset.

	Tech	**Sports**	**Politics**	**Entertainment**	**Business**
Tech	163	3	4	27	10
Sports	3	250	7	1	3
Politics	3	2	178	1	10
Entertainment	1	7	0	173	1
Business	6	2	6	2	250
Accuracy					91.11%

Table 5. Output of the BLSTM-C on the Chinese dataset.

	Tech	**Sports**	**Politics**	**Entertainment**	**Business**
Tech	200	0	0	5	2
Sports	1	258	0	5	0
Politics	2	1	184	1	6
Entertainment	3	5	0	172	2
Business	3	1	4	1	257
Accuracy					96.23%

The tables show that our BLSTM-C model achieves better performance in both experiments, which means that our model is suitable for both Chinese and English languages. It is worth mentioning that our model gets a more significant improvement in the Chinese dataset, a 5.1% higher accuracy than that of a simple LSTM model, while the improvement on the English experiment is only 3.15%.

It can be concluded from the experiments that our BLSTM-C is more suitable for the Chinese language because of the unique structure of Chinese.

5.3. Performance Analysis

Here, the impacts of different parameters on our model performance are analyzed.

- The length of the *maxlen*

  In the initialization and padding of word vectors, the parameter, maxlen, was set up to determine the length of the words that are chosen to represent the article. As for SST-1 and SST-2 datasets, the average length of articles are 18, and the length is so short that it is difficult to obtain the different influences of the article length. Therefore, the THUCNews dataset is selected for an experiment to find out the effect of article length. Figure 4 shows that different lengths of the articles result in different performance.

  The best result, 93.77%, occurs when 80 words are employed to represent the article, which is also deemed as the closest length to the average article length of the dataset. Once the maxlen is far greater than the average length of the articles, then the accuracy will decrease greatly because many more zero vectors will exist in the vectors of the article.

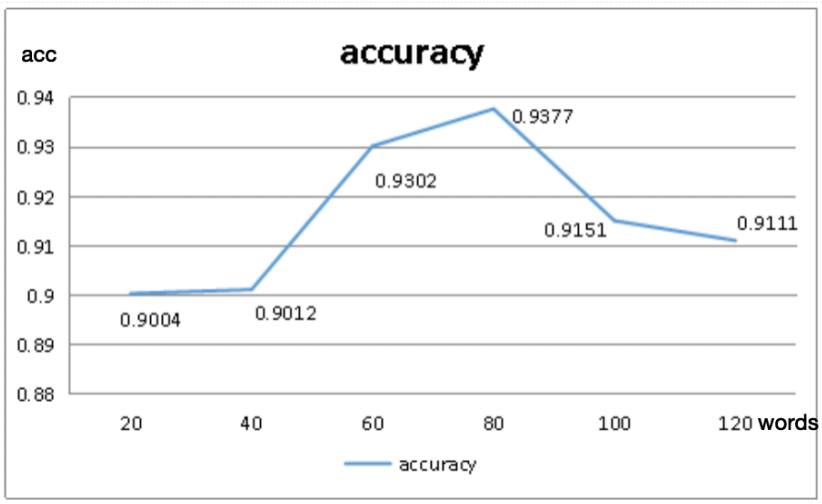

Figure 4. Accuracy vs. article length.

- The size of Convolutional Filter

  In Figure 5, different convolution filter configurations are adopted to present the prediction accuracy on the question classification. As for the horizontal axis, the number indicates convolutional kernel size, and bar charts of five different colors on each filter size represent different dimensions of the convolution output. For example, "S8" means that the size of the kernel is 8 while "F128" denotes that the dimension of the convolution output is 128. It is quite obvious that the dimension of 64 outperforms the other dimensions and the best result, the accuracy of 93.55%, occurs when the window size is 7.

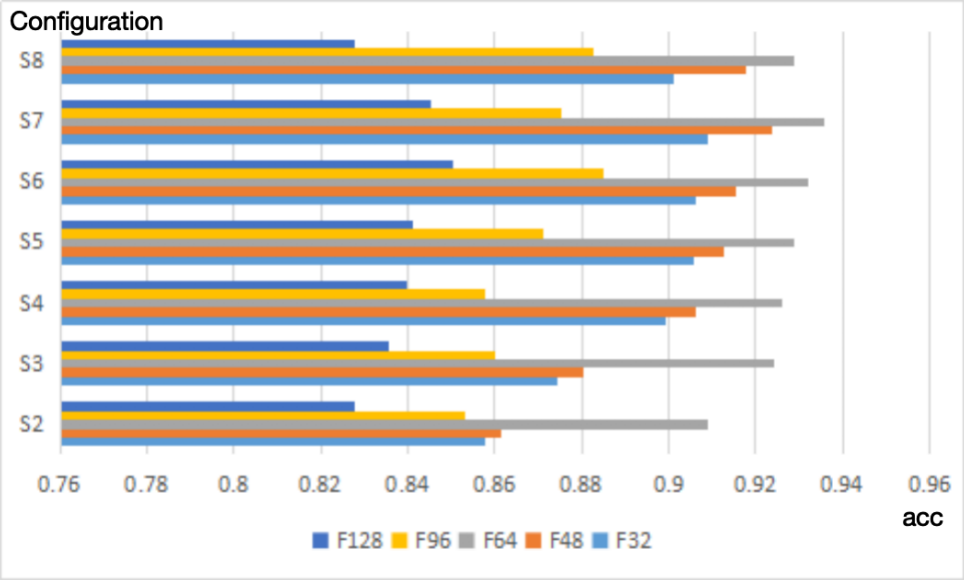

Figure 5. Accuracy vs. different filter configuration.

6. Conclusions

This paper mainly introduces a combined model called BLSTM-C that is made up of a bi-directional LSTM layer and a convolutional layer. The paper also shows its ability in information learning from both previous context and future context, as well as its competence to extract features. It is shown by the experiment results that this model performs remarkably on the tasks related to Chinese News classification, and it also outperforms CNN, RNN and other models on sentiment classification. Moreover, by running the experiments on similar dataset in Chinese and English, it is found out that our BLSTM-C model may have better performance in the Chinese language because the improvement shown in the Chinese language is more significant. To obtain better performance in Chinese news classification, the suitable parameters for this model are also explored and it is found that it is helpful to improve the results by setting the maxlen closest to the average article length and adopting a suitable window to detect the features.

Furthermore, the following suggestions may lead to better performance in future work. Firstly, it will be an interesting idea to deepen the neural network layer. Research on text classification by employing 29 convolutional layers gets satisfying results. Moreover, it is also common for Long Short-term Memory (LSTM) to be multi-layer. Secondly, the article will be more reasonable if every location, number, name and some other meaningless words are replaced with placeholders like '[location]', '[number]','[name]'. This operation will enable the article to be clearer for computer systems.

Therefore, in future work, these modifications will be tried in our experiment to see if it will achieve better performance.

Funding: This research was funded by the Natural Science Foundation of Shanghai Grant No. 16ZR1401100, and the National Key R&D Program of China, Grant No. 2017YFB0309800.

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

References

1. Wang, S.; Manning, C.D. Aselines and bigrams: Simple, good sentiment and topic classification. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, Jeju Island, Korea, 8–14 July 2012; Volume 2, pp. 90–94.
2. Socher, R.; Perelygin, A.; Wu, J.Y.; Chuang, J.; Manning, C.D.; Ng, A.Y.; Potts, C. Recursive deep models for semantic compositionality over a sentiment treebank. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Seattle, WA, USA, 18–21 October 2013.

3. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
4. Cho, K.; Merrienboer, B.V.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv* **2014**, arXiv:1406.1078.
5. Krizhevsky, A.I.S.A.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.
6. Kalchbrenner, N.; Grefenstette, E.; Blunsom, P. A convolutional neural network for modelling sentences. *arXiv* **2014**, arXiv:1404.2188.
7. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–10 December 2013; Volume 2, pp. 3111–3119.
8. Le, Q.; Mikolov, T. Distributed representations of sentences and documents. In Proceedings of the 31st International Conference on Machine Learning (ICML-14), Beijing, China, 21–26 June 2014; pp. 1188–1196.
9. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
10. Kim, Y. Convolutional neural networks for sentence classification. *arXiv* **2014**, arXiv:1408.5882.
11. Conneau, A.; Schwenk, H.; Barrault, L.; Lecun, Y. Very deep convolutional networks for text classification. *arXiv* **2017**, arXiv:1606.01781.
12. Tai, K.S.; Socher, R.; Manning, C.D. Improved semantic representations from tree-structured long short-term memory networks. *arXiv* **2015**, arXiv:1503.00075.
13. 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, Berlin, Germany, 7–12 August 2016; p. 207.
14. Zhang, H.-P.; Yu, H.-K.; Xiong, D.-Y.; Liu, Q. Hhmm-based chinese lexical analyzer ictclas. In Proceedings of the Second SIGHAN Workshop on Chinese Language Processing, SIGHAN '03, Sapporo, Japan, 11–12 July 2003; pp. 184–187.
15. Zhou, C.; Sun, C.; Liu, Z.; Lau, F. A c-lstm neural network for text classification. *arXiv* **2015**, arXiv:1511.08630.
16. Zhu, Z.; Yin, H.; Chai, Y.; Li, Y.; Qi, G. A novel multi-modality image fusion method based on image decomposition and sparse representation. *Inf. Sci.* **2018**, *432*, 516–529. [CrossRef]
17. Lee, K.-F.; Hon, H.-W. Speaker-independent phone recognition using hidden Markov models. *IEEE Trans. Acoust. Speech Signal Process.* **1989**, *37*, 1641–1648. [CrossRef]
18. Socher, R.; Huval, B.; Manning, C.D.; Ng, A.Y. Semantic compositionality through recursive matrixvector spaces. In Proceedings of the Empirical Methods on Natural Language Processing, Jeju Island, Korea, 12–14 July 2012; pp. 1201–1211.
19. Irsoy, O.; Cardie, C. Deep recursive neural networks for compositionality in language. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; pp. 2096–2104.
20. Lei, T.; Barzilay, R.; Jaakkola, T. Molding cnns for text: Non-linear, nonconsecutive convolutions. *arXiv* **2015**, arXiv:1508.04112.
21. Zhu, X.; Sobhani, P.; Guo, H. Long short-term memory over recursive structures. In Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 1604–1612.