INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 13(1), 50-61, February 2018.

Text Classification Research with Attention-based Recurrent Neural Networks

C. Du, L. Huang

Changshun Du*, Lei Huang
School of Economics and Management
Beijing Jiaotong University
Beijing 100044, China
*Corresponding author: summer2015@bjtu.edu.cn

Abstract: Text classification is one of the principal tasks of machine learning. It aims to design proper algorithms to enable computers to extract features and classify texts automatically. In the past, this has been mainly based on the classification of keywords and neural network semantic synthesis classification. The former emphasizes the role of keywords, while the latter focuses on the combination of words between roles. The method proposed in this paper considers the advantages of both methods. It uses an attention mechanism to learn weighting for each word. Under the setting, key words will have a higher weight, and common words will have lower weight. Therefore, the representation of texts not only considers all words, but also pays more attention to key words. Then we feed the feature vector to a softmax classifier. At last, we conduct experiments on two news classification datasets published by NLPCC2014 and Reuters, respectively. The proposed model achieves F-values by 88.5% and 51.8% on the two datasets. The experimental results show that our method outperforms all the traditional baseline systems.
Keywords: machine learning, text classification, attention mechanism, bidirectional RNN, word vector.

1 Introduction

Text classification refers to the process of determining text categories based on the text content under a given classification system. On the Internet, major news sites, forums, blogs and so on are used as text for the main information subject and studying their automatic classification has a wide range of uses. In the field of journalism, press and publication need to be classified according to the columns in order to organize different news in different columns; intelligent recommendation system needs to be calibrated according to the user's different personality characteristics and preferences for the corresponding category of news; in mail processing tasks, the Mail system needs to be governed by the contents of the messages to determine whether the message is spam and decide whether to show to the user. Therefore, the main goal of this paper is to study the text automatic classification algorithm and meet the current mass of automatic classification of text requirements.

As early as the 1960s, people began to study text classification. At that time, it was artificial to write classification rules according to language phenomena and rules. By the 1990s, people began to study computer-based automatic classification technology. This method is first trained by pre-tagging data, learning discrimination rules or classifiers, and then starting to automatically classify new samples of unknown categories. The results show that in the context of large data volumes, its classification accuracy is much better than the expert definition of the rules. Therefore, the current research focuses on automatic text categorization of computer algorithms. Mingzhu Yao [16] et al. used the Latent Dirichlet Allocation (LDA) model to automatically classify text. The LDA model is expressed as a fixed probability distribution, the Gibss sampling

in Markov chain Monte Carlo (MCMC) is used to reason and the parameters of the model are calculated indirectly. The probability distribution of the text on the fixed subject is obtained, the large probability is for the text of the category. Aili Zhang et al. [18] used the support vector machine (SVM) algorithm for multi-class text classification. The method mainly uses the vector space model as a feature, which transforms the document into a high dimension sparse vector per the features of the text and then enters it into the SVM classifier. Liu Hua [4] uses the key phrases of the text to classify, and he thinks that the key words or phrases of the response text category information are more important, so the vector features of the key phrases are first extracted by statistical methods, and then the cosine similarity is calculated to determine the category. With the rise of in-depth learning methods in recent years, the limited Boltzmann machine has also been widely applied to text classification methods. Hilton et al. [10] used the depth Boltzmann machine simulation document to automatically learn the classification characteristics of the document, and in current document classifications has achieved good results. Note that except of document classification, deep learning has been also drawn attention in other pattern classification tasks, such as images [15] [6].

The above methods have achieved some success in text classification tasks, but they each have their own shortcomings. In the literature [4], it is pointed out that the categories of texts are usually associated with some key phrases and words, so they are modeled using the method of extracting keywords. These keywords are important, but other words that link these keywords together also contain a lot of information about the document, and the direct abandonment of these words can seriously damage the information that the document represents. In [10], the neural network is used to study the document, taking into account the interrelations and sequences of words, has the ability to extract text features automatically, and has the strongest performance on current classical data sets. However, the whole model does not take the role of key words into account, but rather treats all the words as a network of input, not giving the key words any special treatment. Therefore, we believe that if we can combine the advantages of the two methods, redesign the neural network model, and increase the weight keywords in the network, that the final text classification results should see a significant improvement. In order to verify this hypothesis, we designed the recursive neural network model to learn the representation of the text, and added the attention mechanism to the neural network [1]. The function of the attention mechanism is to learn a weight for each word of the input document, expecting key words to have a heavier weight, and the non-critical words to have a lighter weight, and the weight of the word reflects its contribution to the subject of the document. In the attention mechanism, the values of these weights are obtained through network learning, which is different from the previous subject model. Here, we will assign a vector to each category. The weight of the word is calculated according to the similarity between the word vector and the category vector. All the vectors in the model, including the word vector and the class vector, are learned through the optimization algorithm.

In the lab section, we collected data sets for news categories in Chinese and English, and experimented with different settings on both datasets. The Chinese data came from The 3rd Natural Language Processing and Chinese Computing Conference (NLPCC2014) public evaluation of news classification data, provided by the Xinhua News Agency and marked category tags into a total of 347 categories; English news data was selected from RCV1-v2 released by Reuters [10], which contains a total of 103 categories. We first use the text preprocessing technique to remove low frequency words and stop words, then use the recursive neural network with the attention mechanism to extract the feature vector of the article, and finally pass the feature vector to the softmax classifier. The experimental results show that the attention mechanism is very effective in assigning a higher weight to the key words, and can effectively improve the accuracy of the classification.

The method proposed in this paper makes full use of the advantages of the depth learning model which can employ self - learning characteristics, and embeds the traditional method of using keywords to classify the text in the neural network by way of the attention mechanism. The advantages of the two are organically combined in this paper. In Part 2, the structure of the model is described in detail. Part 3 gives the optimization objective function of the model and the parameter settings of the experiment. The fourth part shows the experimental results of the model on the above two datasets. Part 5 is the Model and experimental summary.

The development of workable assessment systems is difficult largely due to the fact that the value of assessment is often controversial:

2 Recurrent neural network model based on attention mechanism

The model consists of two parts, the first part is the feature extraction operation which mainly utilizes the recursive neural network to gradually synthesize the vector characteristics of the text. We first introduce the recurrent neural network model, and then describe it in detail on the basis of this model to increase the structure of the attention mechanism; the second part is the classifier, the classifier has a dropout [14] layer and softmax layer composition. The biggest advantage of this model is that only simple preprocessing of text is required, you can use the attention mechanism to select keywords and learn the text of the feature representation. The various parts of the model are described in detail below.

2.1 The representation of the word

In the text, we represent the word as a distributed word vector, and there are already many work studies in the vector field to learn word representations [11] [5] [8] [9] [7]. We use [8] the proposed language model to learn the representation of words. First, we collect an unsupervised text corpus and the New York Times corpus (NYT), pre-training the word vector with the Baidu Encyclopedia. [13] [12] [17] and other work points out that in a large-scale unsupervised corpus, the learning the word vectors can improve the effectiveness of the model, and the model can also provide a better initial value. In this paper, we use $\mathbf{E}$ to represent the matrix of word vectors, each of which represents a vector of words, and the dimension of the column vector is d . The k th word is expressed as a one-hot vector v_k (the k th position is 1 and the remaining position is 0), then the vector of the k th word can be denoted as $\mathbf{E}v_k$.

2.2 The input layer of the network

In the past, the input of the network was the word vector itself. Here, we use this input method as a stepping stone to improve. We believe that the word definition cannot be determined by only the word itself, but also should look at the specific environment where the word is placed, and furthermore see how it works in different environments, where the meaning of the same word differs. This feature is especially important for Chinese context. Therefore, we get the word's pre-training initial value, and the word in each text section, we use it before and after each word as background content, to calculate the given word as the center of the window for the average of the word vector 3, As a vector representation of the current word. Here is an example.

</s> Shanghai Forest Coverage Year by Year Increase </s>

In this sentence, the word "coverage" of the vector is calculated as:

$$V'_{Coverage} = \frac{V_{Forest} + V_{Coverage} + V_{Year}}{3},$$

For the words in the beginning and ending of a text, we use the symbol </s> to fill it, which is also given a vector representation in the model, so the beginning and ending words can also take a similar calculation. It is worth noting that this method of calculation constitutes the input layer of the model, which is included in the objective function expression of the optimization model, not just the initial calculation.

2.3 Recurrent neural network (RNN)

The RNN (Recurrent Neural Network) model has demonstrated a strong learning ability in many natural language processing tasks. It is characterized by good modeling of sequence data and full utilization of sequence information. Since the RNN is to semantically synthesize each word in the text in turn, the RNN can adapt to the variable sentence, that is, the uniformity of the text length is not required, and both long and short texts can be learned.Fig. 1 shows a traditional recurrent neural network structure.

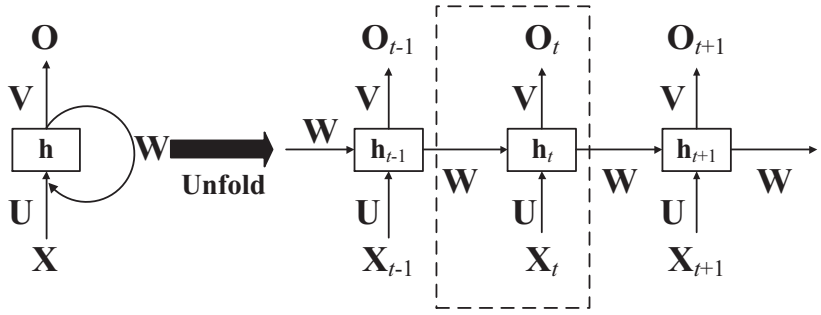


Figure 1: Traditional structure of RNN

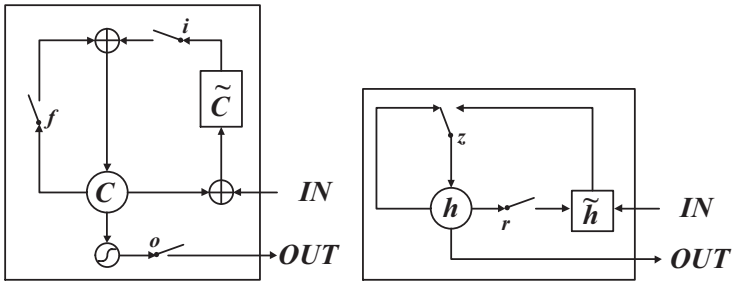
In Fig.1, $\mathbf{x}_t$ is the input unit of step t , which represents the word vector of the t th word in the text; $\mathbf{h}_t$ is the hidden state of step t ; $\mathbf{o}_t$ represents the output of step t , the output of this step is a softmax classifier, the output is selected according to the needs of the model; $\mathbf{U}$,$\mathbf{V}$, and $\mathbf{W}$ are the weight parameters of the network that all need to be learned in the model. As shown in Fig. 1, the dotted line box is the calculation of the t th unit, as follows:

$$\begin{cases} \mathbf{h}_t = f(\mathbf{W}\mathbf{h}_{t-1} + \mathbf{U}\mathbf{x}_t + \mathbf{b}_h) \\ \mathbf{o}_t = softmax(\mathbf{V}\mathbf{h}_t + \mathbf{b}_0) \end{cases} \tag{1}$$

Where variables $\mathbf{b}_h$ and $\mathbf{b}_0$ represent biased terms. As can be seen from Eq.1, each hidden state of the recursive neural network is determined by the current input word and the hidden state of the previous step. If you do not need to add a classifier to each synthetic step in a particular task, $\mathbf{o}_t$ can not be output. The disadvantage of the traditional recurrent neural network is that with the increase of the length of the text, the number of layers of the network is gradually deepened. The loss of the network in the process of information synthesis is relatively large, which tends to focus on the learning of the final stage of memory. Therefore, the efficiency of long text learning is not good.

In this paper, Long Short-term Memory (LSTM) [3] and Gated Recurrent Unit (GRU) [2] are used because of the shortcomings of traditional RNN in dealing with long text. The advantage of the LSTM and GRU nodes is that they can be set up in the process of synthesizing to control how much information should be received in the current synthesis step, how much is forgotten, and how much information is passed back. Through these gate controls, RNN has a proficient learning ability for long text. The difference between LSTM and GRU is that LSTM has more parameters. The GRU has fewer parameters and thus has a faster calculation speed.

LSTM and GRU are two kinds of calculate nodes of RNN. In the method of calculating hidden state, it is different from traditional method, and it is consistent with RNN structure of main body. The LSTM and GRU nodes are calculated as shown in Fig.2.



(a) LSTM structure (b) GRU structure

Figure 2: Structures of LSTM and GRU unit

LSTM node:

$$\begin{cases} \mathbf{i} = \sigma(\mathbf{U}^i\mathbf{x}_t + \mathbf{W}^i\mathbf{h}_{t-1}) \\ \mathbf{f} = \sigma(\mathbf{U}^f\mathbf{x}_t + \mathbf{W}^f\mathbf{h}_{t-1}) \\ \mathbf{o} = \sigma(\mathbf{U}^o\mathbf{x}_t + \mathbf{W}^o\mathbf{h}_{t-1}) \\ \mathbf{g} = tanh(\mathbf{U}^g\mathbf{x}_t + \mathbf{W}^g\mathbf{h}_{t-1}) \\ \mathbf{c}_t = \mathbf{c}_{t-1}o\mathbf{f} + \mathbf{g}o\mathbf{i} \\ \mathbf{h}_t = tanh(\mathbf{c}_t)o\mathbf{o} \end{cases} \tag{2}$$

GRU node:

$$\begin{cases} \mathbf{z} = \sigma(\mathbf{U}^z\mathbf{x}_t + \mathbf{W}^z\mathbf{h}_{t-1}) \\ \mathbf{r} = \sigma(\mathbf{U}^r\mathbf{x}_t + \mathbf{W}^r\mathbf{h}_{t-1}) \\ s = tanh(\mathbf{U}^s\mathbf{x}_t + \mathbf{W}^h(\mathbf{h}_{t-1}o\mathbf{r})) \\ \mathbf{h}_t = (\mathbf{1\text{-}z})o\mathbf{s} + \mathbf{z}o\mathbf{h}_{t-1} \end{cases} \tag{3}$$

σ represents the sigmoid function, and the symbol o represents the operation of the corresponding vector element multiplication. In the LSTM node, $\mathbf{i}$,$\mathbf{f}$, $\mathbf{o}$ represent the input gate, the memory gate, and the output gate respectively, which control the proportion of the information throughput; $\mathbf{g}$ is the hidden state of the candidate, similar to the way the traditional RNN calculates the hidden state; $\mathbf{c}_t$ is the internal memory, by the $t-1$ step of the memory $\mathbf{c}_{t-1}$ and $\mathbf{g}$ through the memory gate and input gate weight to form; $\mathbf{h}_t$ is the true output state, which is the amount of information that the internal memory $\mathbf{c}_t$ outputs at the output gate. In the GRU node, $\mathbf{z}$ is the update gate, $\mathbf{r}$ is the reset gate, s is the hidden state of the current candidate. It can be seen by $\mathbf{s}$ calculation that the reset node controls the amount of the previous node information $\mathbf{h}_{t-1}$, and the final output state $\mathbf{h}_t$ is weighted by the current candidate's hidden state s and the previous node output state $\mathbf{h}_{t-1}$ by updating the gate $\mathbf{z}$.

2.4 Bidirectional RNN model of attention mechanism

In this paper, we use the bidirectional RNN to learn the characteristics of the text, because the meaning of a word is not only related to the text content in front of it, but also related to the text content behind it. We use the bidirectional RNN method to implement the text representated from the learning, and then the two directions to learn the feature vector spliced together, this as a text vector, so that relative to the unidirectional RNN, the eigenvector of the semantics is more comprehensive and rich. At the same time, we add a mechanism of attention to the network model, for each word to learn a weight, making the key words have a heavier

weight, and non-key words have a lighter weight, which can make important features become more prominent. Fig.3 shows the overall architecture of the model.

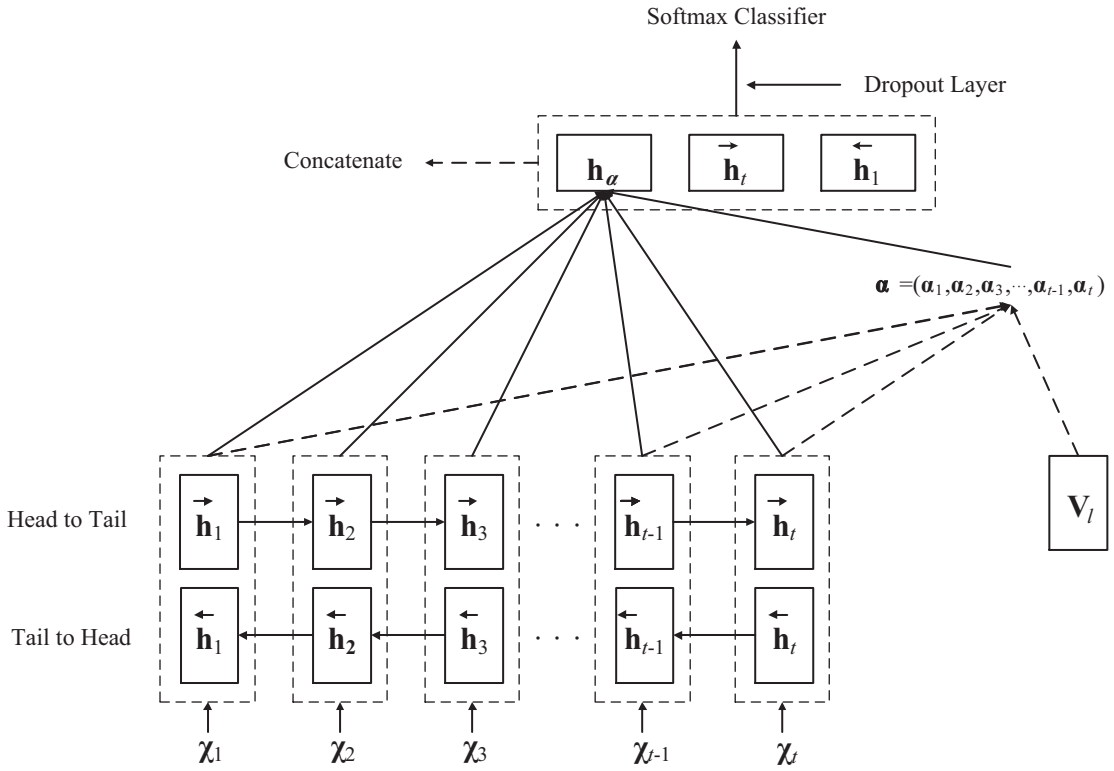


Figure 3: Attention-based bidirectional RNN structure

As shown in Fig.3, the input sentence is x_1 , x_2 , x_3 , L , x_{t-1} , x_t. The recursive neural network RNN has both forward and backward directions, and in the forward RNN, the hidden state is $\overrightarrow{\mathbf{h}_1}, \overrightarrow{\mathbf{h}_2}, \overrightarrow{\mathbf{h}_3}, L, \overrightarrow{\mathbf{h}_{t-1}}, \overrightarrow{\mathbf{h}_t}$ (" $\leftarrow$ " indicates that the direction of the RNN is forward); In this setting, the hidden state of the word x_i corresponds to $\mathbf{h}_i = \left[\overrightarrow{\mathbf{h}_i}; \overleftarrow{\mathbf{h}_i}\right]$ $(i = 1, 2, L, t)$, that is, the hidden state of the two directions together, such as the original hidden state of the k dimensional vector, then the $2k$ vector after the stitching.

In the traditional bidirectional recurrent neural network, the vector of $\overrightarrow{\mathbf{h}_t}$ and $\overleftarrow{\mathbf{h}_1}$ is usually concatenated as a text representation. Since the words that reflect the subject in a text are primarily a few keywords, the importance of each word is different. Here we introduce the attention mechanism. We calculate the corresponding weights for each word by the attention mechanism, and then weighted the sum of the hidden states of all the words according to the weight, and the result of the summation $\mathbf{h}_a$ is also taken as part of the textual feature.

As shown in Fig.3, $\mathbf{v}_l$ represents the category vector, $\alpha_i (i = 1, 2, L, t)$ represents the similarity between the hidden state of the i word and the category vector, that is, the weight of the i word, the similarity formula is [1]

$$\alpha_i = \frac{e^{\mathbf{h}_i^T \mathbf{M} \mathbf{v}_l}}{\sum_{j=1}^{t} e^{\mathbf{h}_i^T \mathbf{M} \mathbf{v}_l}} \tag{4}$$

And $i = 1, 2, L, t$. S is a parameter matrix that calculate generalized similarity, When it is a unit array, the similarity of $\mathbf{h}_i$ and $\mathbf{v}_l$ is degenerated into the inner product. $\alpha = (\alpha_1, \alpha_2, L, \alpha_t)$ represents the weight vector. According to Eq.4 we can see that it has been normalized. The resulting weighted feature vector $\mathbf{h}_a$ is

$$\mathbf{h}_a = \sum_{i=1}^{t} \alpha_i \mathbf{h}_i \tag{5}$$

And then $\mathbf{h}_a$ and RNN forward and backward results together, that is the character representation of the text.

$$\mathbf{s} = \left[\mathbf{h}_a, \overrightarrow{\mathbf{h}_t}, \overleftarrow{\mathbf{h}_1}\right]$$

2.5 Softmax classifier

As shown in Fig.3, after extracting the features of the text, the feature vectors are entered into the softmax classifier for classification. Here we use dropout [14] to connect the feature vector with the softmax classifier. To illustrate the dropout method, we consider the eigenvector as an input to the classifier. The traditional neural network connection method is comprised of the whole connection mode, the dropout algorithm is connected to the random input data(the feature vector after splicing in this paper) according to a certain proportion of 0, and the only other elements that are not set to 0 are ones participating in the operation and connection. For the sake of convenience, suppose that a learning sample is updated once, and the specific process is as follows: First, the input vector is placed in proportion to a portion of the elements, and the element not set to 0 is involved in the operation and optimization of the classifier; then the second sample enters the vector, this time in accordance with the random set way to select the elements of training, until all the samples have been studied once. Since for each input of a sample, the way to set 0 is randomized, each network update will have differing weight parameters. In the final prediction process, the entire network parameters are multiplied by $1 - \rho$ to get the final classifier network parameters. Because the parameters of each update are not the same, the dropout algorithm can be seen as the neural network being cast into a combination of multiple models, and can effectively prevent over-fitting and improve the model's prediction rate [14]. According to the literature [14], the dropout algorithm is similar to evolution, and the genes of the offspring are made up of half of the genes of the parents. This combination has a tendency to produce more vigorous genes. Similarly, in the final network of the dropout algorithm the parameter is a combination of the parameters of multiple models, which is a process of choosing and keeping the advantages over the shortcomings, and thus yielding a better generalization ability.

Suppose that the vector obtained by a bi-directional recurrent neural network is $\mathbf{c}$. The way the dropout algorithm sets its element to 0 can be represented by the Bernoulli distribution. The Bernoulli distribution is used to produce a binary vector $\mathbf{r}$ (it only contains 0 or 1) equal to the latitude of $\mathbf{c}$:

$$\mathbf{r} \sim \mathbf{Bernoulli}\,(\rho)$$

he vector entered into the softmax classifier is recorded as:

$$\mathbf{c}_d = \mathbf{c} \cdot \mathbf{r}$$

Where softmax classifier's network parameter is $\mathbf{W}_c$ and the offset term is $\mathbf{b}_c$, the output of the network is :

$$\mathbf{o} = f\left(\mathbf{W}_c \mathbf{c}_d + \mathbf{b}_c\right)$$

f is a sigmoid function or a tanh function. The probability that the current text belongs to category i is:

$$p\left(i|S\right) = e^{o_i} / \sum_{j=1}^{N} e^{o_j}$$

o_i represents the i th element of the vector $\mathbf{o}$, and N represents the number of classes.

2.6 Objective function

In this paper, we mainly study the classification problem. The parameters that need to be optimized include the following parts: word vectors, parameters of bi-directional recurrent neural networks, generalized similarity matrices of attention mechanisms, class vectors and classifier parameters. A word vector is denoted by $\mathbf{E}$, and the parameters of the bi-directional RNN are denoted by $\hat{\mathbf{W}}$ and $\hat{\mathbf{U}}$; the parameters of the attention mechanism layers are denoted as $\mathbf{M}$, the vector of all the categories is represented by the matrix $\mathbf{V}$, the i line vector $\mathbf{v}_i$ represents the i th category; the parameters of the classifier are denoted by $\mathbf{W}_c$. The sample set of the training set is $\Omega = \left\{\left(T_1, y_1\right), \left(T_2, y_2\right), L, \left(T_{|\Omega|}, y_{|\Omega|}\right)\right\}$, where T_i is the i th text, y_i is its category label, and $|\Omega|$ is the number of training set samples. $\theta = \left\{\mathbf{E}, \hat{\mathbf{W}}, \hat{\mathbf{U}}, \mathbf{M}, \mathbf{V}, \mathbf{W}_c\right\}, p\left(y_i|T_i, \theta\right)$ represents the probability that the category of the text T_i is divided into y_i when the parameter θ is known, so the optimized objective function is:

$$L = \sum_{i=1}^{|\Omega|} logp\left(y_i|T_i, \theta\right) + \lambda \left\|\theta\right\|_2^2$$

λ is the parameter of the regular term. In the actual experiment, we use the random gradient descent method to optimize θ update method is:

$$\theta = \theta - \alpha \frac{\partial L}{\partial \theta},$$

α is the learning rate.

3 Experiments and results

3.1 Experimental data

Two data sets are used in the experiment. The first is the Chinese data set, it is the 2014 Chinese Computer Society organized by the natural language processing conference published by the news classification evaluation data set. It is responsible for organizing and annotating Xinhua, which is a large-scale news classification corpus. It can be downloaded directly from the official website of NLPCC2014. The corpus training set has a size of 30,000 news articles, the test set contains 11,577 articles, and it's test set and training set have good consistency in the distribution of each category. The data set has two categories, the first layer comprised of 24 categories, and the second layer with a total of 367 categories. In this paper, the category of text is unified as a single-level category, and for a multi-level hierarchy of trees, we consider the final small category as a single category and report the classification of the final category. The second data set is an English data set and is an REV1-v2 dataset published by Reuters. This data set contains 804,414 news articles, comprising a total of 103 topics (103 categories, here is the hierarchical classification, processing with the previous data set). Following the literature [10], we randomly divided the data set into a training set and test set, of which the training set contains

794,414 news articles, and the test set contains 1,000 news articles. All of the experiments in this paper are performed on both datasets.

3.2 Data preprocessing

For the Chinese data set, we first use the Chinese word segmentation package NLPIR developed by the Chinese Academy of Sciences for Chinese word segmentation. The functions of NLPIR include Chinese word segmentation, partnered annotation, named entity recognition, user dictionary function, support of a variety of Chinese coding formats, and the ability to discover new words as well as facilitate keyword extraction. As the experiment in this article has a Chinese data set, we need to call the software packet word. English data itself is a separate word, so the operation of word segmentation is unnecessary. After the word segmentation operation is completed, the word frequencies of the two data sets are calculated, and the low frequency words and stop words are deleted. Because these words are not helpful in judging the subject, it may be helpful to use them when classifying them, which is not conducive to the prediction of the classification model.

Since we used the minibatch training model during the training process, we needed to perform a fixed-length operation on the length of the text. Since the sentence lengths of the natural language text are inconsistent, we first calculate the longest sentence length l_{max} . For any sentence length less than l_{max} , the uniform use of the text </s> symbol to l_{max} (</s> vector is always set to $\mathbf{0}$). The purpose of unifying the text length is to improve the efficiency of computing. When the length of the data is unified, you can use a matrix calculation, which when compared with circular computing is time-saving.

3.3 Pre - training of word vector

Before the model training, we need to pre-train the training vector on an unregulated large-scale corpus. A word vector is a distributed representation of a word that expresses an input suitable for a neural network. Many of the current studies have shown that word vectors without oversight learning in a large corpus are more conducive to the convergence of the neural network model leading to a good local optimal solution. In this paper, we use the Skip-gram model to pre-train the training vector. The vector of this model has a strong performance in many natural language processing tasks. The Skip-gram algorithm has been integrated in the word2vec package which we use directly to train Chinese and English word vectors. We use the text content read on the Baidu Encyclopedia to carry out the pre-training of the Chinese word vector, we pre-train the English word vector with the New York Times corpus.

3.4 The setting of the experimental parameters

In this paper, the model has the following super parameters: the dimension of the vector d , the dimension of the class vector, the dimension n of the hidden state in the recurrent neural network, the ratio ρ of the dropout algorithm, and the learning rate α of the SGD optimization algorithm. We use the grid search method to determine these parameters. The dimension of the word vector d is taken in $\{50, 100, 200, 300\}$; The dimension of the class vector l is in $\{50, 100\}$, The dimension n of the hidden state is taken in $\{500, 1000, 2000\}$; According to experience, the dropout algorithm ratio P is 0.6; the SGD algorithm learning rate α is in $\{1, 0.1, 0.01, 0.001\}$. For the Xinhua Newsroom data set, the best parameter value is: $d = 100, l = 100, n = 1000, \alpha = 0.05$. For Reuters RCV1-v2 datasets, the best parameter values are: $d = 300, l = 100, n = 1000, \alpha = 0.01$. The range of these parameters is based on experience, generally within the scope of the value can be achieve better experimental results. In

this experiment, we use these parameters for multiple experiments, and then obtain the average of the results.

3.5 Data experiment and comparative analysis

In this paper, we design a bi-directional recurrent neural network based on an attention mechanism to deal with the problem of text classification. The attention mechanism can be used to learn a weight for each word in the text based on the information of the category, where words closely related to the category receive a relatively heavy weighting, and words that are relatively weak in relation to the category receive lighter weighting. In the experiment, we vectorize 20,000 words of high frequency occurrence in Chinese, and vectorize 100,000 words of high frequency occurrence in an English data set. Tab.1 lists some of the baseline models and the results presented in this paper.

Table 1: Test results of document classification Chinese and English data sets

Model	**Accuracy**		**Recall**		**F-Value**	
	Xinhua News Agency	Reuters corpus RCV1-v2	Xinhua News Agency	Reuters corpus RCV1-v2	Xinhua News Agency	Reuters corpus RCV1-v2
TF-IDF+SVM	72.1	31.8	88.7	45.8	79.5	37.5
AveVec+SVM	70.8	29.3	92.3	33.2	80.1	31.1
TRNN	74.4	40.5	90.7	51.7	81.7	45.4
LDA	77.3	35.1	93.3	44.3	84.5	39.2
DocNADE	76.5	41.7	84.5	42.5	80.3	42.1
Replicated Softmax	82.3	42.1	94.0	47.2	87.8	44.5
Over-Rep. Softmax	82.7	45.3	89.3	51.4	85.9	48.2
Bi-TRNN	82.4	44.8	91.1	52.5	86.5	48.3
LSTM Bi-RNN	81.9	46.2	92.8	55.8	87.0	50.6
GRU Bi-RNN	83.3	45.8	93.9	51.9	88.3	48.7
Attention LSTM Bi-RNN	81.7	46.4	92.8	56.1	87.0	51.8
AttentionGRU Bi-RNN	83.9	46.0	93.5	52.8	88.5	49.2

In order to verify the validity of the model, we compared it with the methods of some baseline systems. The first comparison method is to calculate the TF-IDF feature of the text, form a set of vectors, and then use the support vector machine (SVM) to classify the eigenvectors; The second method is to use the average of the text word vector (the text is preprocessed to calculate the mean of the vector of all words) and then use the SVM classifier for classification; The third method TRNN (Traditional RNN) is to achieve the Traditional RNN model, The fourth method is to call the matlab LDA algorithm package for text classification; The fifth method is the neural autoregressive density estimation method; The sixth and seventh methods are the use of the depth of the Boltzmann built RMB model [10], and the softmax classifier is transformed accordingly; Bi-TRNN is based on the traditional RNN, using the results of a two-way network; Finally, the LSTM Bi-RNN and the GRU Bi-RNN represent the bi-directional recurrent neural networks based on the LSTM and GRU compute nodes, respectively. The Bi-RNN is a bi-directional recurrent neural network. Finally, the Attention LSTM Bi-RNN and Attention GRU

bidirectional Recurrent Neural Network of Attention Mechanism. From the results in Table 1, it can be seen that the neural network model with an attention mechanism achieves the strongest performance on most indicators, reaching 83.9% accuracy and 88.5% F value when tested on the news corpus of Xinhua News Agency. When tested on the Reuters corpus it reached a precision of 46.4% and F value of 51.8% effect. It can be seen that in the task of text learning, the advantages of the two methods of neural network representation learning and traditional keyword classification are taken into account, which has positive significance for the task of text classification.

In addition, in regard to the LSTM and GRU contrast, regardless of whether there is no attention mechanism used in the network structure, because LSTM has more fitting parameters than GRU, it is more suitable for large data learning and prediction. So, for the Reuters news corpus, the predictive effect of LSTM is significantly better than GRU, while for the Xinhua news data, the GRU results are better than the LSTM node.

4 Conclusion

Based on the task of text classification, this paper proposes a bi-directional recurrent neural network algorithm based on the neural network attention mechanism. After extracting the vector features of the text, the feature is input into the softmax classifier by dropout. Previous methods either based on keywords, or the use of neural networks, each have their own shortcomings. The former is too concerned about the keyword and ignores the role of other words. The latter treats all words equally, regardless of the particularity and importance of the keyword. The attention mechanism described in this paper can be a good combination of the advantages of both.

Bibliography

[1] Bahdanau, D.; Kyunghyun Cho, K.; Bengio Y. (2014); Neural machine translation by jointly learning to align and translate, ICLR 2015, *arXiv preprint arXiv*, 1409.0473, 2014.

[2] Chung, J.; Gulcehre, C.; Cho, K. et al. (2015); Gated feedback recurrent neural networks, *International Conference on Machine Learning*, 37, 2067-2075, 2015.

[3] Graves, A.; Schmidhuber, J. (2005); Framewise phoneme classification with bidirectional LSTM and other neural network architectures, *Neural Networks*, 18(5), 602–610, 2005.

[4] Hua, L. (2007); Text Categorization Base on Key Phrases, *Journal of Chinese Information Processing*, 21(4), 34–41, 2007. (in Chinese)

[5] Huang, E.H.; Socher, R.; Manning, C.D.; et al. (2012); Improving word representations via global context and multiple word prototypes, *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, Association for Computational Linguistics, 873–882, 2012.

[6] Li, W.; Wu, G.; Zhang, F.; Du, Q. (2017); Hyperspectral Image Classification Using Deep Pixel-Pair Features, *IEEE Transactions on Geoscience and Remote Sensing*, 55(2), 844-853, 2017.

[7] Luong, T.; Socher, R.; Manning, C.D. (2013); Better Word Representations with Recursive Neural Networks for Morphology, *CoNLL*, 104–113, 2013.

[8] Mikolov, T.; Sutskever, I.; Chen, K.; et al. (2013); Distributed representations of words and phrases and their compositionality, *Advances in neural information processing systems*, 3111–3119, 2013.

[9] Mikolov, T.; Yih, W.T.; Zweig, G. (2013); Linguistic regularities in continuous space word representations, *Proceedings of NAACL HLT 2013*, Atlanta, USA, 746-751, 2013.

[10] Nitish, S.; Salakhutdinov, R.R.; Hinton G.E. (2013); Modeling documents with deep boltzmann machines, *Uncertainty in Artificial Intelligence - Proceedings of the 29th Conference*, 616-624, 2013.

[11] Pennington, J.; Socher, R.; Manning, C.D. (2014); GloVe: Global vectors for word representation, *Conference on Empirical Methods in Natural Language Processing*, Doha, Qatar, 1532–1543, 2014.

[12] Socher, R.; Huval, B.; Manning, C.D.; et al. (2012); Semantic compositionality through recursive matrix-vector spaces, *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Association for Computational Linguistics, 1201–1211, 2012.

[13] Socher, R.; Perelygin, A.; Wu, J.Y.; et al. (2013); Recursive deep models for semantic compositionality over a sentiment treebank, *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, 1631–1642, 2013.

[14] Srivastava, N.; Hinton, G.; Krizhevsky, A.; et al. (2014); Dropout: A simple way to prevent neural networks from overfitting, *The Journal of Machine Learning Research*, 15(1), 1929–1958, 2014.

[15] Xu, X. ; Li, W.; Ran, Q.; et al. (2018); Multisource Remote Sensing Data Classification Based on Convolutional Neural Network, *IEEE Transactions on Geoscience and Remote Sensing*, 56(2), 937-949, 2018.

[16] Yao, Q.Z.; Song, Z.L.; Peng, C. (2011); Research on text categorization based on LDA, *Computer Engineering and Applications*, 47(13), 150–153, 2011. (in Chinese)

[17] Zeng, D.; Liu, K.; Lai, S.; et al. (2014); Relation Classification via Convolutional Deep Neural Network, *COLING*, 2335–2344, 2014.

[18] Zhang, A.-L., Liu, G.-L., Liu C.-Y. (2004); Research on multiple classes text categorization based o SVM, *Journal of Information*, 9, 6–10, 2004. (in Chinese)