IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Text Classification Based on Conditional Reflection

YANLIANG JIN[1], CAN LUO[1], WEISI GUO[2], JINFEI XIE[1], DIJIA WU[1], and RUI WANG[1]

[1]Key laboratory of Specialty Fiber Optics and Optical Access Networks, Joint International Research Laboratory of Specialty Fiber Optics and Advanced Communication, Shanghai Institute for Advanced Communication and Data Science, Shanghai University

[2]School of Engineering, University of Warwick, Coventry, UK; Alan Turing Institute, London, UK

Corresponding author: YanLiang Jin (e-mail: wuhaide@shu.edu.cn).

This work is supported by the NSFC of China Nos. 61771299, the key laboratory of specialty fiber optics and optical access networks, Shanghai University, Nos. SKLSFO2012-14, funding of the key laboratory of wireless sensor network and communication, Shanghai institute of microsystem and information technology, and funding of the Shanghai Education Committee, Chinese academy of sciences and Shanghai science committee Nos. 12511503303, 14511105602 and 14511105902, and H2020 (778305), and Innovate UK (10734).

ABSTRACT Text classification is an essential task in many Natural Language Processing (NLP) applications, we know each sentence may have only a few words that play an important role in text classification, whilst other words have no significant effect on the classification results. Finding these keywords has an important impact on the classification accuracy. In this paper, we propose a network model, named RCNNA (Recurrent Convolution Neural Networks with Attention), which models on the human conditional reflexes for text classification. The model combines bidirectional LSTM (BLSTM), attention mechanism and Convolutional Neural Networks (CNN) as the receptors, nerve centers and effectors in the reflex arc. The receptors get the context information through BLSTM, the nerve centers get the important information of the sentence through the attention mechanism. And the effectors capture more key information by CNN. Finally, the model outputs the classification result by the softmax function. We test our NLP algorithm on four datasets containing Chinese and English for text classification, including a comparison of random initialization word vectors and pre-training word vectors. Experiments show that RCNNA achieves the best performance by comparing with state-of-the-art baseline methods.

INDEX TERMS Attention mechanism, bidirectional LSTM, convolutional neural networks, conditional reflection, text classification.

I. INTRODUCTION

With the popularity of social media, text information is increasing dramatically. The semantic information includes commentaries, news articles, and important information, which may have varying commercial and societal value. Faced with such a large amount of noisy data, this paper proposes a text classification method based on conditioned reflex, which makes it very important in many applications such as web search [1], sentiment analysis, and information extraction [2], [3].

A key problem in text classification is feature representation. Traditional text feature representation is manually defined feature, which is mainly based on the bag-of-words (BoW). Usually, n-grams are represented as text features that represent the relationship between words. In addition, there are other text feature representation methods, such as TF-IDF, MI, pLSA, LDA, etc. [4–7], which are more discriminative features suited to different text lengths and contexts. However, the traditional text feature representation will be very sparse, and the context information or word order will be ignored, and the semantic information of the word cannot be accurately captured, which affects the accuracy of text classification.

With the rapid development of deep learning, the word vectors representation based on neural networks has attracted attention. Word embedding is a distributed representation, which can solve the sparse entity of traditional text representation, and the word vectors can capture syntactic and semantic information.

The CNN for text classification is proposed, which proves that the classification is effective on some datasets [8]. CNN can capture the semantics of text well, However, it is difficult to determine the window size, and small window may lead to the loss of association between words, while large window

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2921976, IEEE Access

may lead to the huge parameter space, that is difficult to train. It is difficult to capture the syntactic and semantic relationship between words. The Recurrent Neural Networks (RNN) can solve this problem [9–11]. The advantage of RNN is that it can capture the context information and the semantics of long texts well. However, the RNN model is a biased model, where the latter words are more dominant than the former ones, which cannot capture the semantics of the whole sentence, because the keyword maybe appear any position. The proposal of attention mechanism can solve this problem [12], [13]. It can give each word a weight to decide how important the word is to the whole sentence, but attention can't capture the deep semantic information.

In order to improve the accuracy of classification task, we propose a network structure (RCNNA) based on conditional reflection for text classification. Reflective arcs include receptors, afferent nerves, nerve centers, efferent nerves, and effectors. Conditional reflex is based on posterior knowledge in the following manner. First, we use pre-training to get the word vectors. And we input the word embedding vectors into the bidirectional LSTM networks [14], because the bidirectional LSTM networks can capture the context information very well, just as we see a sentence, we get the global information at once, which is equivalent to a receptor in conditional reflection. Then we use the attention mechanism as the nerve center to determine which words play a key role in text classification. Finally, we use the CNN and K-max pooling to capture higher-dimensional information for text classification, which is equivalent to an effector [15].

In this paper, we compare our model with the state-of-the-art baseline methods using four datasets, and the classification contains sentiment classification and topic classification.

The main contributions of our work are as follows:

1. We propose RCNNA, an integrated model based on text classification tasks. The model obtains global information and local important information of text to analyze the results of text classification.
2. We imitate human physiological structure of conditioned reflexes to build our network by replacing the receptors, nerve centers, and effectors in conditioned reflex with BLSTM, attention mechanism, and CNN. The text global information is obtained through BLSTM, and each word is weighted by the attention mechanism. Finally, the CNN extract more important feature to obtain the text classification results.
3. The experimental results show that our method achieves the most advanced performance compared to the state-of-the-art baseline methods, and proves that the model structure based on human conditional reflection has better effect on text classification.

The general structure of the paper is as follows. Section II introduces the related work. Section III defines the relevant operators and parts of the network. Section IV shows the experimental results and analysis. Section V concludes our works.

II. RELATED WORK

Over the years, there have been many research methods on text classification. The traditional text classification methods mainly include k-Nearest Neighbor (kNN), Decision Trees, Linear Regression (LR), Naive Bayes (NB), SVM [16], [17], etc. Their feature selection is mainly based on bag-of-words (BoW), n-grams, and TF-IDF. However, these methods all have sparsity problems.

With the development of deep learning, the distributed representation of words solves the sparsity problem [18]. Through the pre-training of word vectors, neural networks have made great progress in text classification. The Convolutional Neural Networks is applied to text classification, which is shallow neural networks [8]. A similar network, just adding the number of convolution layer and using K-max pooling and folding operations for classification, is proposed [15]. In addition to the word-level feature representation, the character-level CNN is firstly applied to classification and promising results are achieved [4]. But the CNN layers in the text classification are relatively shallow. The deep neural network is used for text classification tasks based on character levels, and better result in deeper layers are achieved [19]. The recursive neural network for the text classification task is applied [20]. Due to the gradient vanishing or exploding problem of Recurrent Neural Networks. A tree-structured LSTM networks for text classification is proposed [21]. Subsequently, CNN and LSTM are combined for text classification [22]. A layered network structure is used for sentiment analysis [23], which uses CNN to obtain sentence vectors, and then gets document vectors through bidirectional LSTM for classification. The multi-task training Recurrent Neural Networks is applied for text classification, which improves the generalization of the network [24].

The attention mechanism is proposed and applied to machine translation, using the encder-decoder framework [12]. The attention mechanism is used for the text classification [25]. A hierarchical structure for document classification is proposed [26]. The first layer used attention and LSTM to get the sentence vectors. The second layer used the same structure to get the document vectors and finally for classification.

Unlike these works, we use a network structure similar to human conditioned reflexes for text classification. It can follow same pattern of human learning. We combined BLSTM, attention mechanism and CNN as conditional reflection structures, which achieves excellent results in the four datasets.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2921976, IEEE Access

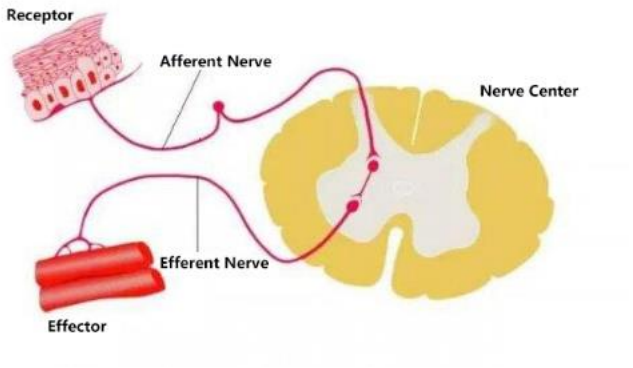

FIGURE 1. **The structure of reflective arc.**

III. MODEL

Our model is based on conditioned reflection. As shown in Figure 1[1]. A reflex arc is a specific neural structure that performs conditioned reflex activity. Information is received from the peripheral receptors, transmitted to the nerve center via the afferent nerve, and the transmitted nerves return the information of the response to the peripheral effector. It is essentially a special contact structure between neurons. The typical pattern generally consists of five parts: receptor, afferent nerve, nerve center, efferent nerve and effector.

In this paper, we use BLSTM, attention mechanism and CNN to represent the receptor, nerve center, and effector, respectively. We know that BLSTM can obtain the entire surrounding information like the receptor, and the information is weighted by the attention mechanism, which is as the nerve center performs a comprehensive analysis of the stimulation of the receptor. Finally, the CNN extracts deeper features to get the corresponding output feedback. As shown in Figure 2, it is the structure of the entire RCNNA networks, which mainly include the following parts, bidirectional LSTM layer, attention mechanism layer, convolution layer and output layer. Each part will be described in the following sections.

A. WORD EMBEDDING

When we input a sentence S, this sentence consists of L words, e.g. $S = \{w_1, w_2, \ldots, w_L\}$, each word is represented by real-valued vectors. For each sentence, we first looking up in the matrix $W \in \mathbb{R}^{d^w|V|}$, where d^w represents the dimension of the word vectors and V represents the vocabulary size. In this paper we use a random vectors matrix or a pre-training matrix trained by word2vec to represent the matrix W. Then each sentence will be represented as word vectors $W_{emb} = \{e_1, e_2, \ldots, e_L\}$.

B. BLSTM LAYER

LSTM was first proposed by Hochreiter and Schmidhuber to solve gradient vanishing problem of RNN [27]. The main idea is to introduce an adaptive gating mechanism. As show in Figure 3, it determines the extent to which the LSTM maintains its previous state and remembers the extracted features, Defining a sentence $S = \{ w_1, w_2, \ldots, w_L \}$, where L represents the length of the sentence, Then the hidden state h_t can be updated with the following equations:

$$i_t = \sigma(W_i w_t + U_i h_{t-1} + b_i) \quad (1)$$

$$f_t = \sigma(W_f w_t + U_f h_{t-1} + b_f) \quad (2)$$

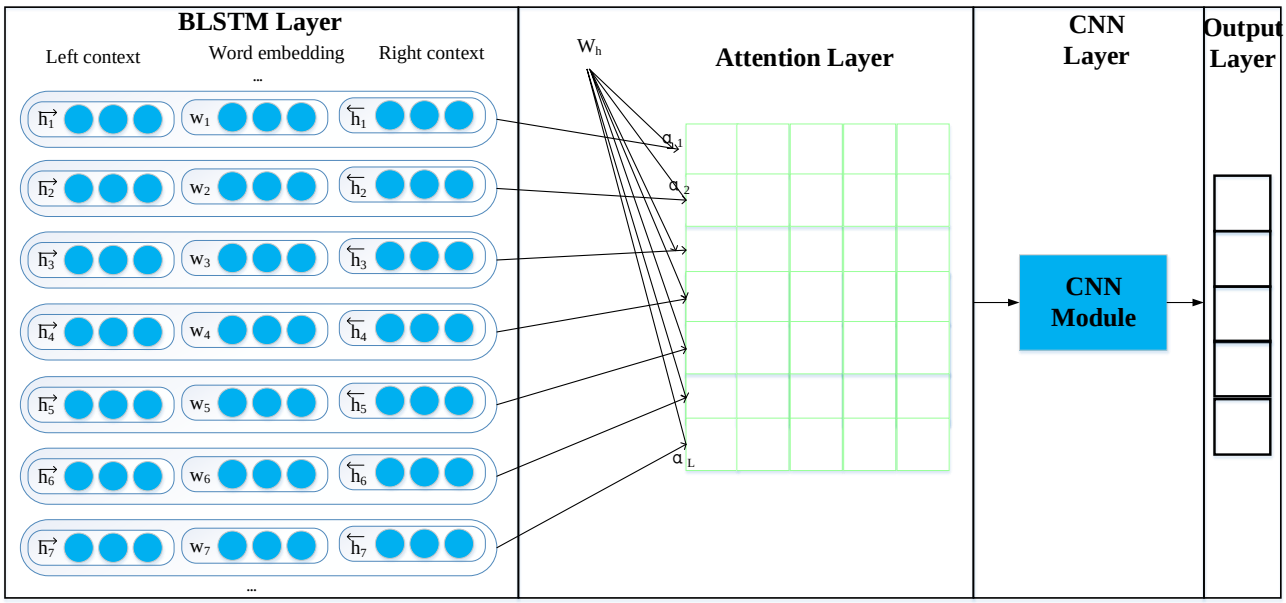

FIGURE 2. **The architecture of RCNNA Network.**

[1]https://ss0.bdstatic.com/70cFuHSh_Q1YnxGkpoWK1HF6hhy/it/u=1063161197,3085858661&fm=26&gp=0.jpg

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2921976, IEEE Access

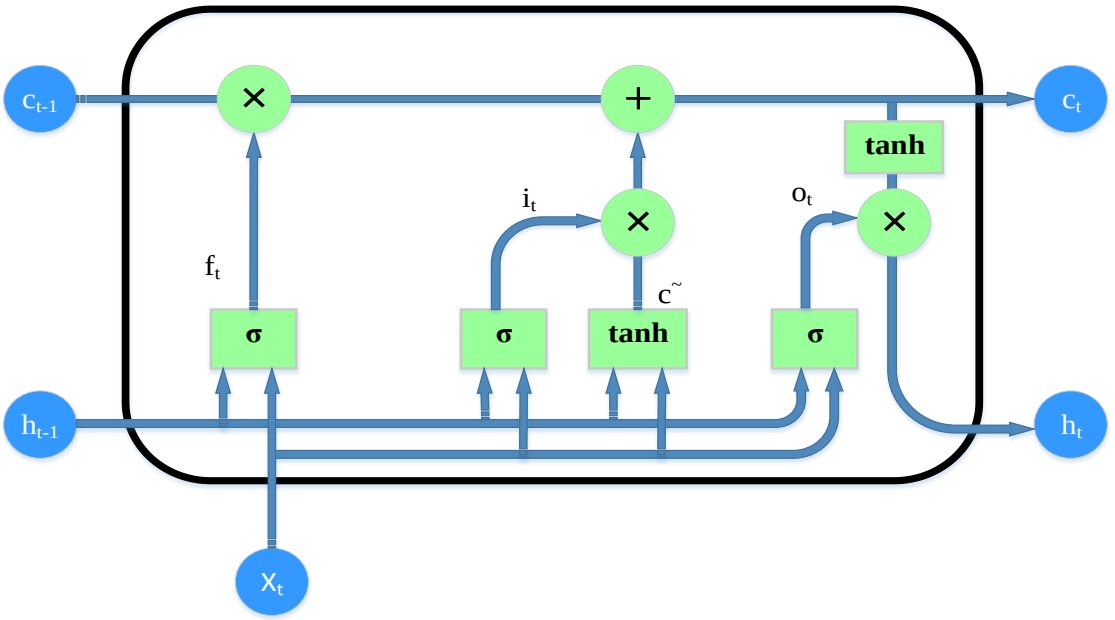

FIGURE 3. **The architecture of LSTM.**

$$o_t = \sigma(W_o w_t + U_o h_{t-1} + b_o) \quad (3)$$

$$\widetilde{c_t} = \tanh(W_c w_t + U_c h_{t-1} + b_c) \quad (4)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \widetilde{c_t} \quad (5)$$

$$h_t = o_t \odot \tanh(c_t) \quad (6)$$

Where i_t, f_t, and o_t are input gate, forget gate and output gate. The parameters W_i , U_i , b_i are the weight matrix of the input gate. The parameters W_f, U_f, b_f are the weight matrix of forget gate. The parameters W_o, U_o, b_o are the weight matrix of the output gate. The parameters W_c , U_c, b_c are the weight matrix of new memory content $\widetilde{c_t}$. w_t is the input of the current time step, c_t is the current cell state, σ represents the logical sigmoid function, and $\odot$ denotes element-wise multiplication.

In the sequence modeling task, Schuster and Paliwal proposed a two-way LSTM by extending an LSTM and flowing in the opposite direction [28]. This can take advantage of past and future information.

In this paper, we used a bidirectional LSTM structure to get context information for sentences. In addition to, we connect the context information and the word vectors to represent the output vectors of the i^{th} word. This enables a close relationship between words. As shown in Figure 2. As such, the output vectors of the i^{th} word is as follows:

$$x_i = [\overrightarrow{h_i} \oplus w_i \oplus \overleftarrow{h_i}] \quad (7)$$

where $\oplus$ represents the connection symbol.

C. ATTENTION MECHANISM

When we are seeing a picture or a piece of text, we always pay attention to the more important information. Given a sentence $S = \{w_1, w_2, \ldots, w_L\}$, we know that the contribution of each word to a text classification is different in the sentence. In this paper, we propose the double attention mechanism for text classification task, let X be a matrix consisting of output vectors $[x_1, x_2, \ldots, x_L]$ of the BLSTM layer. We first use a fully connection to get the hidden layer vectors M, then we define a w_h vectors to represent the importance of the real vectors of each word. We multiply the vectors M and the obtained hidden layer vectors one by one, and average each word vectors, then we use the softmax function for the entire sentence, so we get the weight of each word $[\alpha_1, \alpha_2, \ldots, \alpha_L]$.

$$M = \tanh(W_w X + b_w) \quad (8)$$

$$\alpha = \mathrm{softmax}(\mathrm{average}(w_h \odot M)) \quad (9)$$

$$R = \alpha \odot X \quad (10)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2921976, IEEE Access

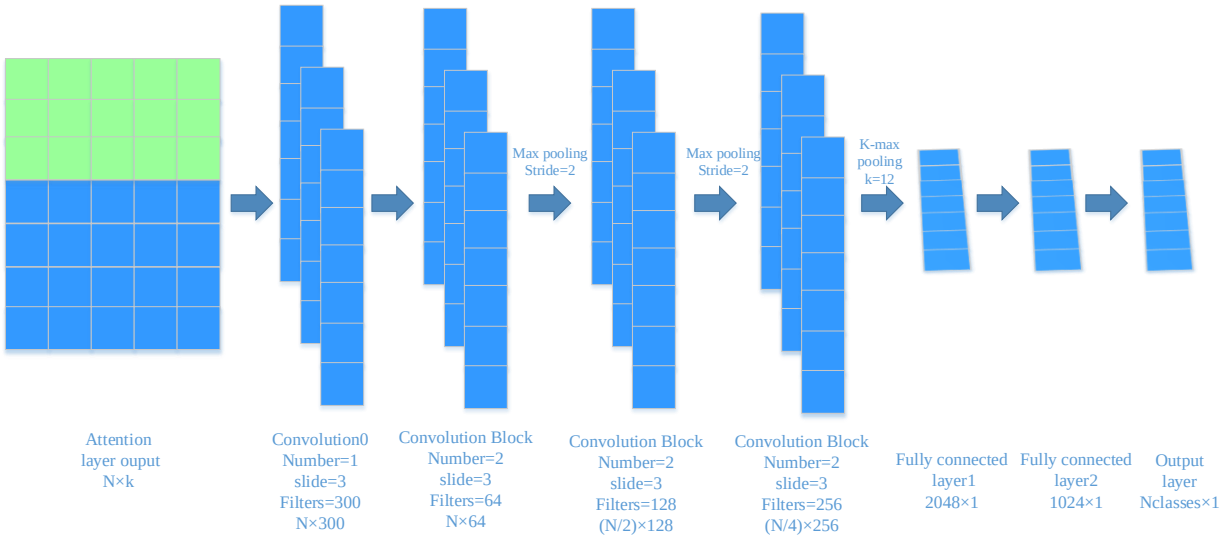

FIGURE 4. **The architecture of CNN Module.**

Where $M \in \mathbb{R}^{d^x \times L}$, d^x is the dimension of the sum of the word vectors and the context vectors, W_w, b_w are the weight matrix, and $R \in \mathbb{R}^{d^x \times L}$, the dimension of α is L, $\odot$ denotes element-wise multiplication.

D. CONVOLUTIONAL NEURAL NETWORKS

Through the attention mechanism we can get important information from a picture or a piece of text. The Convolutional Neural Networks can help us capture more meaningful information. In this paper, the entire Convolutional Neural Networks structure is shown in Figure 4. We know that the VGG networks has a good effect on image classification [29]. In this paper, we refer to the network structure of VGG, but it is not exactly the same. We first convolve the output vectors of the attention to get the first layer convolution vectors, then go through there convolution blocks, each of which contains two convolution sub-layers, and each one contains the Batch Normalization layer and ReLU activation function. We set the convolution sliding window is 3 in this paper. Between the convolutional blocks, we use max-pooling for dimensionality reduction, and the stride=2. Each pooling layer is halved. In the final convolution, we used K-max pooling operation. The value of k varies according to the length of the sentence. Then we use the two-layer fully connected network to get the penultimate layer vectors h^*.

E. OUTPUT LAYER

In this section, we use the softmax function to predict the label $\hat{y}$ from the real category label Y for a sentence S. We use the penultimate layer output h^* as the input to the output layer:

$$\hat{p}(y/S) = \text{softmax}(W^S h^* + b^S) \tag{11}$$

$$\hat{y} = \arg\max_y \hat{p}(y/S) \tag{12}$$

The cost loss function that we use is cross-entropy loss, and we still use L2 regularization, the specific formula is as follows:

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m} t_i \log(y_i) + \lambda||\theta||^2 \tag{13}$$

where $t \in \mathbb{R}^m$ is the one-hot represented ground truth, $y \in \mathbb{R}^m$ is the estimated probability for each class by softmax function, m is the number of categories, and λ is a L2 regularization hyper-parameter, training is done through stochastic gradient descent over shuffled mini-batches with the Adam update rule [30].

IV. EXPERIMENTS

A. DATASETS

In order to prove the validity of our proposed method, we perform experiments on four datasets: Movie Reviews (MR), DBpedia (DB), Hotel Comment (HC), Sina Comment (SC). Table I provides the detailed information for each dataset.

- MR: Movie reviews with one sentence per review. Classification involves detecting positive/negative reviews [31].
- Hotel Comment: The hotel reviews data from the Ctrip website, a Chinese dataset containing positive and negative comments.
- DBpedia: This dataset was created by selecting 14 non-overlapping classes from DBpedia 2014, including Company, Education, Institution, etc.
- Sina Comment: This data set is from Chinese commentary data on Sina Weibo, including 4 categories of joy, anger, disgust, and low.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2921976, IEEE Access

TABLE I
SUMMARY STATISTICS FOR THE DATASETS

Data	C	Train	Dev	Test	Len	V	$\|V_{pre}\|$	Lang
MR	2	9596	-	CV	20	18765	16448	EN
HC	2	5420	622	1657	86	21638	17056	CH
DBpedia	14	558663	55866	69853	89	563348	76336	EN
SC	4	361745	-	CV	91	309700	147311	CH

C: number of target classes, Train/Dev/Test: Train/Development/Test set size, Len: average sentence length, |V|: vocabulary size, $|V_{pre}|$: number of words present in the set of pre-training word embeddings, CV: 10-fold cross validation, Lang: English, Chinese.

B. EXPERIMENT SETTINGS

Throughout the experiment, we used the NVIDIA GEFORCE 1050TI to train our model. For the MR and HC datasets, we use one day of training time, and the remaining two datasets use three days of training time. For the four datasets. We use the following methods for processing. For English data, we use NLTK tool [2] to segment each sentence. For Chinese documents, we use the jieba[3] word segmentation tool for word segmentation. The stop-word list is not used for all datasets. In the Chinese word segmentation, we removed the irregular symbols. The four datasets are then divided into training sets and test sets. The two datasets HC and DBpedia have defined training sets, verification sets, and test sets. For the remaining two datasets, we randomly used 90% of the dataset as the training set, and the remaining 10% as the test set. All experiments use accuracy as a metric.

The setting for the hyper-parameter depends on the dataset used. In general, we set the learning rate l=0.0001, the dimension of the word vectors is 300, and the hidden layer unit of BLSTM is 300. We use the dropout operation to set the BLSTM layer with dropout rate 0.5, we use the one-dimensional convolution slip. The window size is 3, the pool size is 2, except that the length of the word in the MR dataset is L=56, and the other datasets are set L=100. We choose the regularization parameter λ=0.001, in addition, in the MR dataset, we only use the max-pooling once, because the average word length of the MR dataset is very short. For each data set, we can fine-tuning the network to achieve better training results. Batch size is set to 32 in the MR dataset and 64 in the other datasets. We use random initialization word vectors and pre-training word vectors. For the English data set, we use the word vectors trained by word2vec on 100 billion words from Google News [32]. For the Chinese dataset, we use the word vectors trained by word2vec on Baidu Encyclopedia data. Words that are not in the pre-training word set are initialized randomly with a uniform distribution of [-0.25, 0.25].

TABLE II
THE RESULTS OF DIFFERENT BASELINE METHODS AND OUR MODEL

Model	Embedding&rand				Embedding&pre-training			
	MR	**HC**	**DB**	**SC**	**MR**	**HC**	**DB**	**SC**
CNN-word (Kim,2014)	0.761	0.865	0.982	0.578	0.815	0.868	0.983	0.621
CNN-char (Zhang et al.2016)	0.748	-	0.973	-	-	-	-	-
Fast-CNN (Miklov et al.2016)	0.730	0.858	0.965	0.596	0.771	0.869	0.976	0.611
RNN-word (Liu et al.2016)	0.770	0.844	0.977	0.627	0.798	0.872	0.980	0.640
RCNN-word (Lai et al.2015)	0.786	0.867	0.978	0.630	0.804	0.879	0.980	0.654
Att+LSTM (Yang et al.2016)	0.788	0.859	0.976	0.624	0.806	0.877	0.983	0.650
VDCNN (Conneau et al.2017)	0.721	0.839	0.970	0.554	0.766	0.863	0.980	0.643
RCNNA	**0.796**	**0.873**	**0.985**	**0.635**	**0.820**	**0.890**	**0.988**	**0.658**

[2] http://www.nltk.org/

[3] https://pypi.org/project/jieba/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2921976, IEEE Access

C. COMPARRISION OF METHODS

We compare our approach with the methods widely used for text classification on each dataset.

CNN-word word-based convolutional neural networks use shallow neural networks to classify text using only convolution, for example [8].

CNN-char is based on character level convolutional neural networks for the first time for text classification [4].

Fast-CNN is a simple improvement based on word2vec as text classification [33].

RNN-word is based on the text classification model of RNN, we chose for comparison [24].

RCNN-word combines CNN as a text classification based on LSTM [22].

Attention+LSTM used the attention mechanism based on LSTM achieve good results in text classification [26].

VDCNN-char deep convolutional neural networks is used for text classification [19].

D. RESULTS AND ANALYSIS

The results of all datasets are shown in Table II: whether it is a Chinese dataset or an English dataset, we have achieved the best results regardless of the size of the datasets when using random word vectors matrices. We achieve 79.6% and 87.3% accuracy on the small datasets MR and HC. On the other two big datasets DB and SC, we also achieve the best accuracy, 98.5% and 63.5%.

From Table II we can see that when we use pre-training word vectors as training, all the accuracy rates on the four datasets are almost higher than those using random initialized word vectors, which means that the certain prior knowledge can improve the accuracy rate.

In addition, we compare the baseline methods CNN and RNN. The baseline methods of RNN are almost higher than CNN on the four datasets. Especially on MR dataset, VDCNN is in random initialization word vectors and the pre-training word vectors achieves 72.1% and 76.6% accuracy, which is the lowest in all the entire baseline methods because MR is a small dataset and the deep convolutional neural networks may present over-fitting. Compared with Fast-CNN, CNN-word, CNN-char methods, RCNN-word has almost some certain improvement on the four datasets. It is indicated that the features captured by CNN only stays in the syntax and sentence information of the text, and the RNN can further capture the semantic feature information. For this reason, we suggest that the multi-layer BLSTM network structure can be used to obtain deeper semantic information. The introduction of attention mechanism has further improved the accuracy of text classification. The Attention+LSTM baseline method achieve an accuracy of 80.6% on the MR dataset, especially on DBpedia dataset, the accuracy is very high, because the dataset is non-overlapping data, and the noise is very small.

Comparing with these results, we find that RCNNA based on human conditional emission can more effectively obtain text global information and make judgments on important words, which will better understand a sentence or text and achieve better classification effect.

V. CONCLUSION

In this paper, we propose a new model (RCNNA) for text classification task. The model imitates human conditional reflection to build a network, which does not depend on the size of the datasets and completely modeled on the human learning model. The experimental results show that our model can obtain important information of the sentence and achieved the best performance by comparing other state-of-the-art baseline methods. In the future work, we will consider different network structures via neuroevolutionary meta-learning to replace the various parts of the conditioned reflection to achieve a better bionic learning model.

REFERENCES

[1] T. G. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms," *Neural Comput.*, vol. 10, no. 7, pp. 1895–1923, Oct. 1998.

[2] C. C. Aggarwal and C.-X. Zhai, "A survey of text classification algorithms," in *Mining Text Data*. New York, NY, USA: Springer, 2012.

[3] Z. Jianqiang, G. Xiaolin, and Z. Xuejun, "Deep convolution neural networks for twitter sentiment analysis," *IEEE Access*, vol. 6, pp. 23253–23260, 2018.

[4] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in neural information processing systems*, 2015, pp. 649–657.

[5] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.

[6] L. Cai and T. Hofmann, "Text categorization by boosting automatically extracted concepts," in P*roc. 26th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2003, pp. 182–189.

[7] H. Swapnil, C. Sandeep, P. Girish K, "Document classification by topic labeling," *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2013: 877– 880.

[8] Y. Kim, "Convolutional neural networks for sentence classification," arXiv Preprint, arXiv:1408.5882, 2014.

[9] R. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Computa*., vol. 1, pp. 270–280.

[10] T. Mikolov, M. Karafiat, L. Burget, J. Černocký, and S. Khudanpur. Recurrent neural network based language model. In *INTERSPEECH*, 2010. 2, 4.

[11] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017.

[12] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv*:1409.0473, 2014.

[13] R. Lin, S. Liu, M. Yang, M. Li, M. Zhou, and S. Li, "Hierarchical recurrent neural network for document modeling," in *Proc. Conf. Empirical Methods Natural Lang. Process*., 2015, pp. 899–907.

[14] D. Zhang and D. Wang, "Relation Classification via Recurrent Neural Network," *CoRR*, vol. abs/1508.01006, 2015.

[15]] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," in *Proc. 52nd Annu. Meeting Association Computational Linguistics*, 2014, vol. 1, pp. 655–665.

[16] S. Keerthi, S. Shevade, C. Bhattacharyya, and K. Murthy, "Improvements to Platt's SMO algorithm for SVM classifier design," *Neural Computation*, vol. 13, pp. 637–649, Mar. 2001.

[17] J. Zhou, Y. Yang, S. X. Ding, Y. Zi, and M. Wei, "A Fault Detection and Health Monitoring Scheme for Ship Propulsion Systems Using SVM Technique," *IEEE Access*, vol. 6, pp. 16207-16215, 2018.

[18] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A Neural Probabilistic Language Model," *J. Machine Learning Research*, vol. 3, pp. 137-1155, 2003.
[19] A. Conneau, H. Schwenk, L. Barrault, and Y. LeCun. (Jun. 2016). "Very deep convolutional networks for text classification." [Online]. Available: https://arxiv.org/abs/1606.01781.
[20] R. Socher et al., "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank," Proc. Empirical Methods on Natural Language Processing, 2013, pp. 1642–1654.
[21] K. S. Tai, R. Socher, and C. D. Manning, "Improved semantic representations from tree-structured long short-term memory networks," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics 7th Int. Joint Conf. Natural Language Processing*, vol. 1, Jul. 2015, pp. 1556–1566, http://www.aclweb.org/anthology/P15-1150.
[22] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification." in *AAAI*, vol. 333, 2015, pp. 2267–2273.
[23] Duyu Tang, Bing Qin, and Ting Liu, "Document modeling with gated recurrent neural network for sentiment classification," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp.1422–1432.
[24] P. Liu, X. Qiu, and X. Huang, "Recurrent neural network for text classification with multi-task learning," in *Proc. Conf. Artif. Intell.*, 2016, pp. 2873–2879.
[25] P. Zhou, Z. Qi, S. Zheng, J. Xu, H. Bao, and B. Xu, "Text classification improved by integrating bidirectional lstm with two-dimensional max pooling," *arXiv preprint arXiv*:1611.06639, 2016.
[26] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proc. NAACL-HLT*, San Diego, CA, USA, 2016, pp. 1480–1489.
[27] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735– 1780, 1997.
[28] M. Schuster and K. K. Paliwal, "Bidirectional Recurrent Neural Networks," *IEEE Transactions on Signal Processing*, vol. 45, pp. 2673–2681, 1997.
[29] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv*:1409.1556, 2014.
[30] KINGMA, D., AND BA, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv*:1412.6980 (2014).
[31] B. Pang and L. Lee, "Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales," *Proc. 43rd Ann. Meeting of the Assoc. for Computational Linguistics (ACL '05)*, 2005.
[32] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Int. Conf. Learn. Representations*, 2013.
[33] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov. "Bag of tricks for efficient text classification." *arXiv preprint arXiv*:1607.01759, 2016.