Densely Connected Bidirectional LSTM with Applications to Sentence Classification

Zixiang Ding[1], Rui Xia[1(✉)], Jianfei Yu[2], Xiang Li[1], and Jian Yang[1]

[1] School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China
{dingzixiang,rxia,xiang.li.implus,csjyang}@njust.edu.cn

[2] School of Information Systems, Singapore Management University, Singapore, Singapore
jfyu.2014@phdis.smu.edu.sg

Abstract. Deep neural networks have recently been shown to achieve highly competitive performance in many computer vision tasks due to their abilities of exploring in a much larger hypothesis space. However, since most deep architectures like stacked RNNs tend to suffer from the vanishing-gradient and overfitting problems, their effects are still understudied in many NLP tasks. Inspired by this, we propose a novel multi-layer RNN model called densely connected bidirectional long short-term memory (DC-Bi-LSTM) in this paper, which essentially represents each layer by the concatenation of its hidden state and all preceding layers hidden states, followed by recursively passing each layers representation to all subsequent layers. We evaluate our proposed model on five benchmark datasets of sentence classification. DC-Bi-LSTM with depth up to 20 can be successfully trained and obtain significant improvements over the traditional Bi-LSTM with the same or even fewer parameters. Moreover, our model has promising performance compared with the state-of-the-art approaches.

Keywords: Sentence classification · Densely connected
Stacked RNNs

1 Introduction

With the recent trend of deep learning, various kinds of deep neural architectures have been proposed for many tasks in speech recognition [2], computer vision [13] and natural language processing (NLP) [6], which have been shown to achieve better performance than both traditional methods and shallow architectures. However, since conventional deep architectures often suffer from the well-known vanishing-gradient and overfitting problems, most of them are not easy to train and therefore cannot achieve very satisfactory performance.

To address these problems, different approaches have been recently proposed for various computer vision tasks, including Highway Networks [16], ResNet [3]

M. Zhang et al. (Eds.): NLPCC 2018, LNAI 11109, pp. 278–287, 2018.
https://doi.org/10.1007/978-3-319-99501-4_24

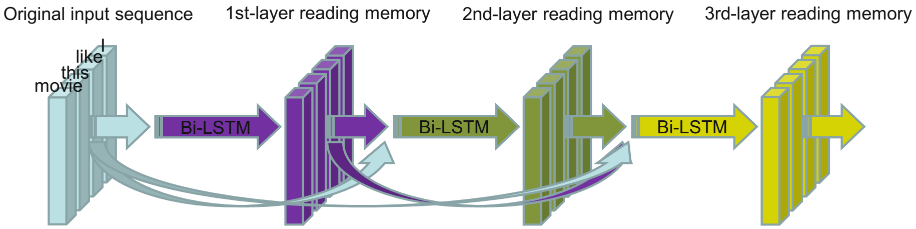

Fig. 1. The architecture of DC-Bi-LSTM. We obtain first-layer reading memory based on original input sequence, and second-layer reading memory based on the position-aligned concatenation of original input sequence and first-layer reading memory, and so on. Finally, we get the n-th-layer reading memory and take it as the final feature representation for classification.

and GoogLeNet [17,18]. One of the representative work among them is the recently proposed Dense Convolutional Networks (DenseNet) [5]. Different from previous work, to strengthen information flow between layers and reduce the number of parameters, DenseNet proposes to directly connect all layers in a feed-forward fashion and encourages feature reuse through representing each layer by concatenating the feature-maps of all preceding layers as input. Owing to this well-designed densely connected architecture, DenseNet obtains significant improvements over the state-of-the-art results on four highly competitive object recognition benchmark tasks (CIFAR-10, CIFAR-100, SVHN, and ImageNet).

Motivated by these successes in computer vision, some deep architectures have also been recently applied in many NLP applications. Since recurrent neural networks (RNNs) are effective to capture the flexible context information contained in texts, most of these deep models are based on the variants of RNNs. Specifically, on basis of Highway Networks, Zhang *et al.* [23] proposed Highway LSTM to extend stacked LSTM by introducing gated direct connections between memory cells in adjacent layers. Inspired by ResNet, Yu *et al.* [21] further proposed a hierarchical LSTM enhanced by residual learning for relation detection task. However, to the best of our knowledge, the application of DenseNet to RNN has not been explored in any NLP task before, which is the motivation of our work.

Therefore, in this paper, we propose a novel multi-layer RNN model called Densely Connected Bidirectional LSTM (DC-Bi-LSTM) for sentence classification. In DC-Bi-LSTM, we use Bi-LSTM to encode the input sequence, and regard the sequence of hidden states as reading memory for each layer. The architecture of DC-Bi-LSTM is shown in Fig. 1. We evaluate our proposed architecture on five sentence classification datasets, including Movie Review Data [11] and Stanford Sentiment Tree-bank [15] for fine-grained and polarity sentiment classifications, TREC dataset [9] for question type classification and subjectivity classification dataset [10]. DC-Bi-LSTM with depth up to 20 can be successfully trained and significantly outperform the traditional Bi-LSTM with the same or even fewer parameters. Moreover, our model achieves indistinguishable performance in comparison with the state-of-the-art approaches.

2 Model

In this section, we describe the architecture of our proposed Densely Connected Bidirectional LSTM (DC-Bi-LSTM) model for sentence classification.

2.1 Deep Stacked Bi-LSTM

Given an arbitrary-length input sentence $S = \{w_1, w_2, \ldots, w_s\}$, Long Short-Term Memory (LSTM) [4] computes the hidden states $h = \{h_1, h_2, \ldots, h_s\}$ by iterating the following equations:

$$h_t = \text{lstm}(h_{t-1}, e(w_t)). \tag{1}$$

where $e(w_t) \in R^m$ is the word embedding of w_t.

As shown in Fig. 2(a), deep stacked Bi-LSTM [1,14] uses multiple Bi-LSTMs with different parameters in a stacking way. The hidden state of l-layer Bi-LSTM can be represented as h_t^l , which is the concatenation of forward hidden state $\overrightarrow{h_t^l}$

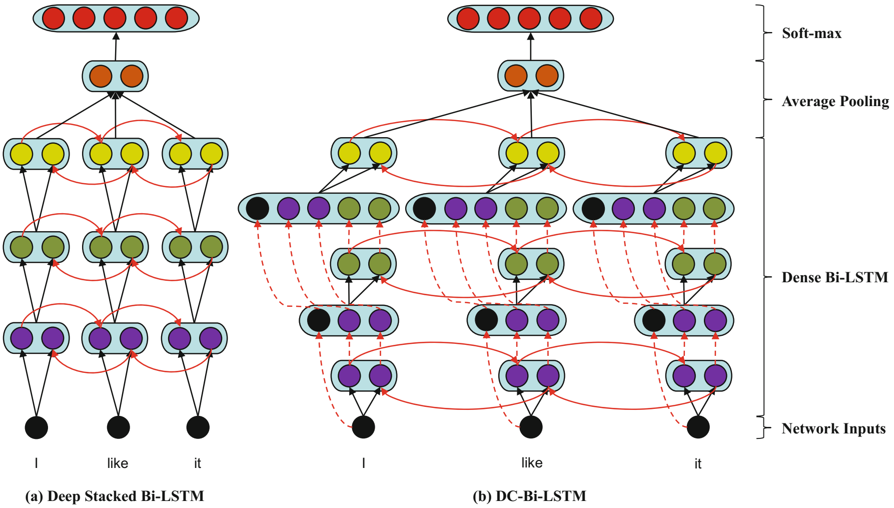

Fig. 2. Illustration of (a) Deep Stacked Bi-LSTM and (b) DC-Bi-LSTM. Each black node denotes an input layer. Purple, green, and yellow nodes denote hidden layers. Orange nodes denote average pooling of forward or backward hidden layers. Each red node denotes a class. Ellipse represents the concatenation of its internal nodes. Solid lines denote the connections of two layers. Finally, dotted lines indicate the operation of copying. (Color figure online)

and backward hidden state $\overleftarrow{h_t^l}$. The calculation of h_t^l is as follows:

$$h_t^l = [\overrightarrow{h_t^l}; \overleftarrow{h_t^l}], \text{ specially, } h_t^0 = e(w_t), \tag{2}$$

$$\overrightarrow{h_t^l} = \text{lstm}(\overrightarrow{h_{t-1}^l}, h_t^{l-1}), \tag{3}$$

$$\overleftarrow{h_t^l} = \text{lstm}(\overleftarrow{h_{t+1}^l}, h_t^{l-1}). \tag{4}$$

2.2 Densely Connected Bi-LSTM

As shown in Fig. 2(b), Densely Connected Bi-LSTM (DC-Bi-LSTM) consists of four modules: network inputs, dense Bi-LSTM, average pooling and soft-max layer.

(1) Network Inputs

The input of our model is a variable-length sentence, which can be represented as $S = \{w_1, w_2, \ldots, w_s\}$. Like other deep learning models, each word is represented as a dense vector extracted from a word embedding matrix. Finally, a sequence of word vectors $\{e(w_1), e(w_2), \ldots, e(w_s)\}$ is sent to the dense Bi-LSTM module as inputs.

(2) Dense Bi-LSTM

This module consists of multiple Bi-LSTM layers. For the first Bi-LSTM layer, the input is a word vector sequence $\{e(w_1), e(w_2), \ldots, e(w_s)\}$, and the output is $h^1 = \{h_1^1, h_2^1, \ldots, h_s^1\}$, in which $h_t^1 = [\overrightarrow{h_t^1}; \overleftarrow{h_t^1}]$ as described in Sect. 3.2. For the second Bi-LSTM layer, the input is not the sequence $\{h_1^1, h_2^1, \ldots, h_s^1\}$ (the way stacked RNNs use), but the concatenation of all previous outputs, formulated as $\{[e(w_1); h_1^1], [e(w_2); h_2^1], \ldots, [e(w_s); h_s^1]\}$, and the output is $h^2 = \{h_1^2, h_2^2, \ldots, h_s^2\}$. For the third layer, whose input is $\{[e(w_1); h_1^1; h_1^2], [e(w_2); h_2^1; h_2^2], \ldots, [e(w_s); h_s^1; h_s^2]\}$, like the second layer does. The rest layers process similarly and omitted for brevity. The above process is formulated as follows:

$$h_t^l = [\overrightarrow{h_t^l}; \overleftarrow{h_t^l}], \text{ specially, } h_t^0 = e(w_t), \tag{5}$$

$$\overrightarrow{h_t^l} = \text{lstm}(\overrightarrow{h_{t-1}^l}, M_t^{l-1}), \tag{6}$$

$$\overleftarrow{h_t^l} = \text{lstm}(\overleftarrow{h_{t+1}^l}, M_t^{l-1}), \tag{7}$$

$$M_t^{l-1} = [h_t^0; h_t^1; ...; h_t^{l-1}]. \tag{8}$$

(3) Average Pooling

For a L layer Dense Bi-LSTM, the output is $h^L = \{h_1^L, h_2^L, \ldots, h_s^L\}$. Average pooling module reads in h^L and calculate the average value of these vectors, the computation can be formulated as $h^* = \text{average}(h_1^L, h_2^L, \ldots, h_s^L)$.

(4) Soft-max Layer

This module is a simple soft-max classifier, which takes h^* as features and generates predicted probability distribution over all sentence labels.

2.3 Potential Application Scenario

From a semantic perspective, the dense Bi-LSTM module adds multi-read context information of each word into their original word vector in a concatenation way: h^1 is the first reading memory based on the input sentence S, h^2 is the second reading memory based on S and h^1, h^k is the k-th reading memory based on S and all previous reading memory. Since the word vector for each word is completely preserved, this module is harmless and can be easily added to other models that use RNN. For example, in the task of machine translation and dialog system, the Bi-LSTM encoder can be replaced by dense Bi-LSTM module and may bring improvements.

3 Experiments

3.1 Dataset

DC-Bi-LSTM are evaluated on several benchmark datasets. Movie Review Data(MR) is a popular sentiment classification dataset proposed by Pang and Lee 2005 [11]. Stanford Sentiment Treebank(SST-1) is an extension of MR [15]. And each review has fine-grained labels, moreover, phrase-level annotations on all inner nodes are provided. SST-2 is the same dataset as SST-1 but used in binary mode without neutral sentences. Subjectivity dataset(Subj) is from Pang and Lee 2004 [10], where the task is to classify a sentence as being subjective or objective. TREC is a dataset for question type classification task [9]. The sentences are questions from 6 classes.

3.2 Implementation Details

In the experiments, we use publicly available 300-dimensional Glove vectors, the number of hidden units of top Bi-LSTM (the last Bi-LSTM layer in dense Bi-LSTM module) is 100, for the rest layers of dense Bi-LSTM module, the number of hidden units and layers are 13 and 15 respectively.

For training details, we use the stochastic gradient descent (SGD) algorithm and Adam update rule with shuffled mini-batch. Batch size and learning rate are set to 200 and 0.005, respectively. As for regularization, dropout is applied for word embeddings and the output of average pooling, besides, we perform L2 constraints over the soft-max parameters.

3.3 Results

Results of DC-Bi-LSTM and other state-of-the-art models on five benchmark datasets are listed in Table 1. Performance is measured in accuracy. We can see that DC-Bi-LSTM gets consistently better results over other methods, specifically, DC-Bi-LSTM achieves new state-of-the-art results on three datasets (MR, SST-2 and Subj) and slightly lower accuracy than BLSTM-2DCNN on TREC and SST-1. In addition, we have the following observations:

- Although DC-Bi-LSTM is a simple sequence model, but it defeats Recursive Neural Networks models and Tree-LSTM, which relies on parsers to build tree-structured neural models.
- DC-Bi-LSTM obtains significant improvement over the counterparts (Bi-LSTM) and variant (LR-Bi-LSTM) that uses linguistic resources.
- DC-Bi-LSTM defeats all CNN models in all datasets.

Above observations demonstrate that DC-Bi-LSTM is quite effective compared with other models.

Table 1. Classification accuracy of DC-Bi-LSTM against other state-of-the-art models. The best result of each dataset is highlighted in **bold**. There are mainly five blocks: (i) traditional machine learning methods; (ii) Recursive Neural Networks models; (iii) Recurrent Neural Networks models; (iv) Convolutional Neural Net-works models; v) a collection of other models. **SVM**: Support Vector Machines with unigram features [15] **NB**: Na-ive Bayes with unigram features [15] **Standard-RNN**: Standard Recursive Neural Network [15] **RNTN**: Recursive Neural Tensor Network [15] **DRNN**: Deep Recursive Neural Network [6] **LSTM**: Standard Long Short-Term Memory Network [19] **Bi-LSTM**: Bidirectional LSTM [19] **Tree-LSTM**: Tree-Structured LSTM [19] **LR-Bi-LSTM**: Bidirectional LSTM with linguistically regularization [12] **CNN-MC**: Convolutional Neural Network with two channels [8] **DCNN**: Dynamic Convolutional Neural Network with k-max pooling [7] **MVCNN**: Multi-channel Variable-Size Convolution Neural Network [20] **DSCNN**: Dependency Sensitive Convolutional Neural Networks that use CNN to obtain the sentence representation based on the context representations from LSTM [22] **BLSTM-2DCNN**: Bidirectional LSTM with Two-dimensional Max Pooling [24].

Model	MR	SST-1	SST-2	Subj	TREC
SVM [15]	-	40.7	79.4	-	-
NB [15]	-	41.0	81.8	-	-
Standard-RNN [15]	-	43.2	82.4	-	-
RNTN [15]	-	45.7	85.4	-	-
DRNN [6]	-	49.8	86.6	-	-
LSTM [19]	-	46.4	84.9	-	-
Bi-LSTM [19]	81.8	49.1	87.5	93.0	93.6
Tree-LSTM [19]	-	51.0	88.0	-	-
LR-Bi-LSTM [12]	82.1	50.6	-	-	-
CNN-MC [8]	81.1	47.4	88.1	93.2	92.2
DCNN [7]	-	48.5	86.8	-	93.0
MVCNN [20]	-	49.6	89.4	93.9	-
DSCNN [22]	81.5	49.7	89.1	93.2	95.4
BLSTM-2DCNN [24]	82.3	**52.4**	89.5	94.0	**96.1**
DC-Bi-LSTM (**ours**)	**82.8**	51.9	**89.7**	**94.5**	95.6

3.4 Discussions

Moreover, we conducted some experiments to further explore DC-Bi-LSTM. For simplicity, we denote the number of hidden units of top Bi-LSTM (the last Bi-LSTM layer in dense Bi-LSTM module) as *th* , for the rest layers of dense Bi-LSTM module, the number of hidden units and layers are denoted as *dh* and *dl* respectively. We tried several variants of DC-Bi-LSTM with different *dh*, *dl* and *th*, The results are shown below.

(1) Better parameter efficiency

Better parameter efficiency means obtaining better performance with equal or fewer parameters. In order to verify DC-Bi-LSTM has better parameter efficiency than Bi-LSTM, we limit the number of parameters of all models at 1.44 million (1.44M) and conduct experiments on SST-1 and SST-2. The results are shown in Table 2.

Table 2. Classification accuracy of DC-Bi-LSTM with different hyper parameters. We limit the parameters of all models at 1.44M in order to verify DC-Bi-LSTM models have better parameter efficiency than Bi-LSTM.

dl	*dh*	*th*	Params	SST-1	SST-2
0	10	300	1.44M	49.2	87.2
5	40	100	1.44M	49.6	88.4
10	20	100	1.44M	51.0	88.5
15	13	100	1.40M	**51.9**	**89.7**
20	10	100	1.44M	50.2	88.8

The first model in Table 2 is actually Bi-LSTM with 300 hidden units, which is used as the baseline model, and the results are consistent with the paper [19]. Based on the results of Table 2, we get the following conclusions:

- DC-Bi-LSTM improves parameter efficiency. Pay attention to the second to the fifth model, compared with baseline model, the increase on SST-1(SST-2) are 0.4% (1.2%), 1.8% (1.3%), 2.7% (2.5%) and 1% (1.6%), respectively, with the parameters not increased, which demonstrates that DC-Bi-LSTM models have better parameter efficiency than base-line model
- DC-Bi-LSTM models are easy to train even when the they are very deep. We can see that DC-Bi-LSTM with depth of 20 (the fifth model in Table 3) can be successfully trained and gets better results than baseline model. In contrast, we trained deep stacked LSTM on SST-1, when depth reached more than five, the performance (For example, 30% when the depth is 8, which drops 19.2% compared with baseline model) drastically decreased.
- The fifth model performs worse than the fourth model, which indicates that too many layers will bring side effects when limiting the number of

parameters. One possible reason is that more layer lead to fewer hidden units (to ensure the same number of parameters), impairing the ability of each Bi-LSTM layer to capture contextual information.

(2) Effects of increasing depth (*dl*)

In order to verify that increasing *dl* does improve performance of DC-Bi-LSTM models, we increase *dl* gradually and fix *dh* at 10 . The results on SST-1 and SST-2 are shown in Table 3.

Table 3. Classification accuracy of DC-Bi-LSTM with different hyper parameters. We increase *dl* gradually and fix *dh* at 10 in order to verify that increasing *dl* does improve performance of DC-Bi-LSTM models.

dl	*dh*	*th*	Params	SST-1	SST-2
0	10	100	0.32M	48.5	87.5
5	10	100	0.54M	49.4	88.1
10	10	100	0.80M	49.5	88.4
15	10	100	1.10M	**50.6**	**88.8**
20	10	100	1.44M	50.2	**88.8**

The first model in Table 3 is actually Bi-LSTM with 100 hidden units, which is used as the baseline model. Based on the results of Table 3, we can get the following conclusions:

- It is obvious that the performance of DC-Bi-LSTM is positively related to *dl*. Compared with baseline model, DC-Bi-LSTM with *dl* equal to 5, 10, 15 and 20 get improvements on SST-1 (SST-2) by 0.9% (0.6%), 1.0% (0.9%), 2.1% (1.3%) and 1.7% (1.3%) respectively.
- Among all models, the model with *dl* equal to 15 works best. As *dl* continues to increase, the accuracy does not further improve, nevertheless, there is no significant decrease.

(3) Effects of adding hidden units (*dh*)

In this part, we explore the effect of *dh*. The number of layers in dense Bi-LSTM module (*dl*) is fixed at 10 while the number of hidden units (*dh*) is gradually increased. The results on SST-1 and SST-2 are shown in Table 4. Similarly, we use Bi-LSTM with 100 hidden units as baseline model (the first model in Table 4). Based on the results of Table 4, we can get the following conclusions:

- Comparing the first two models, we find that the second model outperforms baseline by 0.7% on SST-1 and 0.8% on SST-2, which shows that even if *dh* is equal to 5, DC-Bi-LSTM are still effective.
- As *dh* increases, the performance of DC-Bi-LSTM steadily increases. One possible reason is that the ability of each layer to capture contextual information is enhanced, which eventually leads to the improvement of classification accuracy.

Table 4. Classification accuracy of DC-Bi-LSTM with different hyper parameters. We increase *dh* gradually and fix *dl* at 10 in order to explore the effect of *dh* models.

dl	*dh*	*th*	Params	SST-1	SST-2
10	0	100	0.32M	48.5	87.5
10	5	100	0.54M	49.2	88.3
10	10	100	0.80M	49.5	88.4
10	15	100	1.10M	50.2	88.4
10	20	100	1.44M	**51.0**	**88.5**

4 Conclusion and Future Work

In this work, we propose a novel multi-layer RNN model called Densely Connected Bidirectional LSTM (DC-Bi-LSTM) for sentence classification tasks. DC-Bi-LSTM alleviates the problems of vanishing-gradient and overfitting and can be successfully trained when the networks are as deep as dozens of layers. We evaluate our proposed model on five benchmark datasets of sentence classification, experiments show that our model obtains significant improvements over the traditional Bi-LSTM and gets promising performance in comparison with the state-of-the-art approaches. As future work, we plan to apply DC-Bi-LSTM in the task of machine translation and dialog system to further improve their performance, for example, replace the Bi-LSTM encoder with dense Bi-LSTM module.

Acknowledgments. The work was supported by the Natural Science Foundation of China (No. 61672288), and the Natural Science Foundation of Jiangsu Province for Excellent Young Scholars (No. BK20160085).

References

1. El Hihi, S., Bengio, Y.: Hierarchical recurrent neural networks for long-term dependencies. In: NIPS (1996)
2. Graves, A., Mohamed, A.r., Hinton, G.: Speech recognition with deep recurrent neural networks. In: ICASSP (2013)
3. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
4. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
5. Huang, G., Liu, Z., Weinberger, K.Q., van der Maaten, L.: Densely connected convolutional networks. In: CVPR (2017)
6. Irsoy, O., Cardie, C.: Deep recursive neural networks for compositionality in language. In: NIPS (2014)
7. Kalchbrenner, N., Grefenstette, E., Blunsom, P.: A convolutional neural network for modelling sentences. arXiv preprint arXiv:1404.2188 (2014)
8. Kim, Y.: Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882 (2014)

9. Li, X., Roth, D.: Learning question classifiers. In: COLING (2002)
10. Pang, B., Lee, L.: A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In: ACL (2004)
11. Pang, B., Lee, L.: Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales. In: ACL (2005)
12. Qian, Q., Huang, M., Lei, J., Zhu, X.: Linguistically regularized LSTMs for sentiment classification. arXiv preprint arXiv:1611.03949 (2016)
13. Russakovsky, O., et al.: Imagenet large scale visual recognition challenge. Int. J. Comput. Vis. **115**(3), 211–252 (2015)
14. Schmidhuber, J.: Learning complex, extended sequences using the principle of history compression. Neural Comput. **4**(2), 234–242 (1992)
15. Socher, R., et al.: Recursive deep models for semantic compositionality over a sentiment treebank. In: EMNLP (2013)
16. Srivastava, R.K., Greff, K., Schmidhuber, J.: Training very deep networks. In: NIPS (2015)
17. Szegedy, C., et al.: Going deeper with convolutions. In: CVPR (2015)
18. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: CVPR (2016)
19. Tai, K.S., Socher, R., Manning, C.D.: Improved semantic representations from tree-structured long short-term memory networks. arXiv preprint arXiv:1503.00075 (2015)
20. Yin, W., Schütze, H.: Multichannel variable-size convolution for sentence classification. arXiv preprint arXiv:1603.04513 (2016)
21. Yu, M., Yin, W., Hasan, K.S., dos Santos, C., Xiang, B., Zhou, B.: Improved neural relation detection for knowledge base question answering. arXiv preprint arXiv:1704.06194 (2017)
22. Zhang, R., Lee, H., Radev, D.: Dependency sensitive convolutional neural networks for modeling sentences and documents. arXiv preprint arXiv:1611.02361 (2016)
23. Zhang, Y., Chen, G., Yu, D., Yaco, K., Khudanpur, S., Glass, J.: Highway long short-term memory RNNs for distant speech recognition. In: ICASSP (2016)
24. Zhou, P., Qi, Z., Zheng, S., Xu, J., Bao, H., Xu, B.: Text classification improved by integrating bidirectional LSTM with two-dimensional max pooling. arXiv preprint arXiv:1611.06639 (2016)