Integrating Bidirectional LSTM with Inception for Text Classification

Wei Jiang, Zhong Jin
School of Computer Science and Engineering
University of Science & Technology
Nanjing, People's Republic of China
e-mail: jiangw@njust.edu.cn, zhongjin@njust.edu.cn

***Abstract*—A novel neural network architecture, BLSTM-Inception v1, is proposed for text classification. It mainly consists of the BLSTM-Inception module, which has two parts, and a global max pooling layer. In the first part, forward and backward sequences of hidden states of BLSTM are concatenated as double channels, rather than added as single channel. The second part contains parallel asymmetric convolutions of different scales to extract nonlinear features of multi-granular n-gram phrases from double channels. The global max pooling is used to convert variable-length text into a fixed-length vector. The proposed architecture achieves excellent results on four text classification tasks, including sentiment classifications, subjectivity classification, and especially improves nearly 1.5% on sentence polarity dataset from Pang and Lee compared to BLSTM-2DCNN.**

Keywords-Text classification; BLSTM; Parallel asymmetric convolution; Global max pooling

I. INTRODUCTION

Neural network methods have recently achieved great success in the field of text classification [4,5,7,25], and significantly outperform the traditional methods that are based on bag-of-words (BoW) model which is short of encoding word order and syntactic information. Actually, distributed representations of words [11] that are also called word embeddings are the key to breakthrough of deep learning in Natural Language Process (NLP) field. Because it provides dense and low dimensional real vector representations of words which can capture rich semantic and syntactic properties that make it possible to use various advanced neural architectures like convolutional neural networks (CNNs) [4] and recurrent neural networks (RNNs) [9,19] to model complex structure of sentences and documents.

CNNs and RNNs have emerged as two mainstream neural network architectures in the NLP filed and are often combined with sequence or tree structure [12,19], such as simple CNN model used by Kim [4] and Tree-LSTM model proposed by Tai et al. [19]. The convolution operation in CNNs can effectively and efficiently capture local correlations of spatial or temporal structure. So it is often applied in text classification to extract the features of n-gram phrases. However, CNNs are short of extracting the global features of the sentence or document like long-term dependencies. Owing to the recurrent structure, it is natural to apply RNNs to process the variable-length text. And the proposed Long Short-Term Memory Network (LSTM) can effectively capture long-term dependencies by relieving the problem of gradient exploding or vanishing in standard RNNs. Recently, bidirectional LSTM (BLSTM) has also been a baseline in text classification like the simple CNN used by Kim [4]. However, BLSTM does not perform well in extracting local features of the sentence or document. So it is a good idea to combine CNNs and RNNs to utilize their advantages of extracting local and global features of text at the same time. Actually, much work has sprung up in this research direction recently [7,24,25]. For examples, Zhou et al. [24] proposed a simple combined architecture named C-LSTM which utilizes CNN to extract a window sequence of features of n-gram phrases that is then fed into LSTM to obtain the sentence representation. Liang and Zhang [7] replaced the convolution in above architecture with asymmetric convolutions. Zhou et al. [25] introduced two-dimensional convolution and max pooling to the hidden states sequence came from BLSTM.

The proposed architecture BLSTM-Inception v1 mainly bases on BLSTM-2DCNN model proposed by Zhou et al. [25] and Inception v3 [18]. Inspired by Inception modules in [18], BLSTM-Inception module is proposed by integrating BLSTM with asymmetric convolutions of different scales. Firstly, the forward and backward outputs from BLSTM [22] are concatenated as two channels better than added as single channel used in BLSTM-2DCNN. Besides that, ordinary convolutional layer used in BLSTM-2DCNN is replaced with three parallel asymmetric convolutional layers each of which consists of two successive asymmetric convolutions where the second convolution is similar to wide convolution. Instead of flattening the output of CNN layer as in BLSTM-2DCNN, global max pooling is used after BLSTM-Inception module that is demonstrated to greatly improve the generalization. Finally, the proposed BLSTM-Inception v1 architecture achieves excellent results on 4 of 5 text classification tasks.

II. MODEL

As shown in Fig.1, the whole neural network architecture consists of embedding layer, BLSTM-Inception module, global max pooling layer and softmax layer. The core of the framework is the BLSTM-Inception module which consists of BLSTM and three parallel asymmetric convolutional layers with different filter width. The detailed description of these components will be depicted in the following sections.

2327-0985/17 $31.00 © 2017 IEEE
DOI 10.1109/ACPR.2017.113

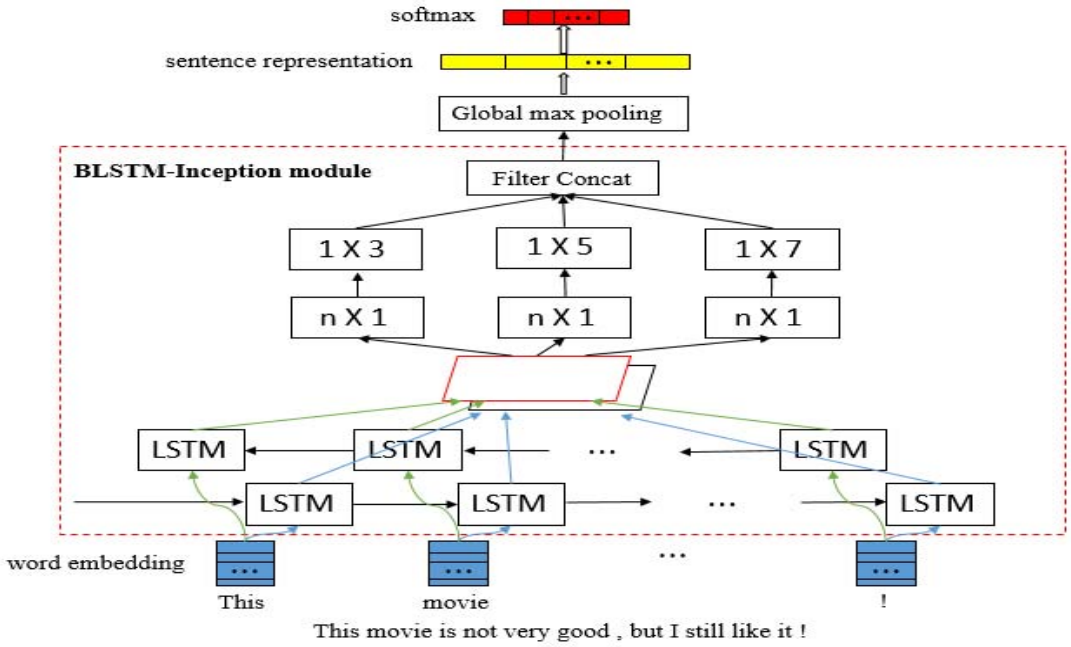

Figure 1: BLSTM-Inception v1 architecture for an example sentence

A. BLSTM Part in BLSTM-Inception Module

Let $x_t \in \mathbb{R}^d$ be the word embedding whose dimensionality is d corresponding to the t-th word in the sequence of words that could be a sentence or document whose length is l, then it can be represented as a matrix $S = [x_1, \cdots, x_l] \in \mathbb{R}^{d \times l}$. The LSTM variant used in this paper is depicted in Zaremba et al. [22] in which the size of units in LSTM equals to the dimensionality of the word embedding. Its dynamics can be described by using deterministic transition as following:

$$\text{LSTM}: x_t, h_{t-1}, c_{t-1} \to h_t, c_t \tag{1}$$

$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} sigm \\ sigm \\ sigm \\ tanh \end{pmatrix} T_{2d,4n} \begin{pmatrix} x_t \\ h_{t-1} \end{pmatrix} \tag{2}$$

$$c_t = f \odot c_{t-1} + i \odot g \tag{3}$$

$$h_t = o \odot \tanh(c_t) \tag{4}$$

where $h_t, c_t \in \mathbb{R}^n$ are respectively hidden state and memory state at time step t and initialized with zero at $t = 0$, $sigm$ and $tanh$ are respectively sigmoid and hyperbolic tangent activation functions, i, f, o, g are respectively input gate , forget gate , output gate and new candidate memory state, $\odot$ denotes element-wise multiplication , $T_{2d,4n}: \mathbb{R}^{2d} \to \mathbb{R}^{4n}$ is an affine transform ($Wx + b$ for some W and b).

The bidirectional LSTM contains the forward part $\overrightarrow{\text{LSTM}}$ which reads the sequence S from x_1 to x_l and the backward part $\overleftarrow{\text{LSTM}}$ which reads from x_l to x_1, and two parts share the parameters:

$$\overrightarrow{h_t}, \overrightarrow{c_t} = \overrightarrow{\text{LSTM}}\left(x_t, \overrightarrow{h_{t-1}}, \overrightarrow{c_{t-1}}\right), t \in \{1, \cdots, l\} \tag{5}$$

$$\overleftarrow{h_{l-t+1}}, \overleftarrow{c_{l-t+1}} = \overleftarrow{\text{LSTM}}\left(x_t, \overleftarrow{h_{l-t}}, \overleftarrow{c_{l-t}}\right), t \in \{l, \cdots, 1\} \tag{6}$$

Finally, the forward hidden states matrix $\overrightarrow{H} = [\overrightarrow{h_1}, \cdots, \overrightarrow{h_l}]$ and the backward hidden states matrix $\overleftarrow{H} = [\overleftarrow{h_l}, \cdots, \overleftarrow{h_1}]$ can be viewed as two channels of the overall representation $H = [\overrightarrow{H}, \overleftarrow{H}] \in \mathbb{R}^{n \times l \times 2}$ as shown in Fig.1.

B. Inception Part in BLSTM-Inception Module

This part takes H as input. Let w be the width of convolution filter, then the size of convolution which is often used in text classification [4,20] is $n \times w$. In this paper, it will be factorized into two successive convolution operations with $n \times 1$ and $1 \times w$ sizes respectively. The factorization will increase the depth of the network which makes it possible to extract nonlinear features compared to single convolution [8] and decrease the parameters of the network which can relieve over-fitting [18]. The detailed definition of asymmetric convolution layer in which the second convolution is different from [7] will be given in the following part.

The first convolution in asymmetric convolution layer with $n \times 1$ size filter $W_j^1 \in \mathbb{R}^{n \times 1 \times 2}$ is applied to the input patch H_i which stands for the i-th slice along times step dimensionality, that is $H(:, i, :)$, then a corresponding local feature is generated as following:

$$f_{ij}^1 = \max(W_j^1 * H_i + b, 0) \tag{7}$$

where subscript $j \in \{1, \cdots, m\}$ is used to index the channels of the feature map and superscript 1 denoted the first convolution, b is a bias term and nonlinear activation ReLU is chosen. Then, the feature maps we got are $f^1 \in \mathbb{R}^{1 \times l \times m}$ which satisfies

$$f^1(1, :, :) = (f_{ij}^1)_{l \times m} \tag{8}$$

The second convolution of asymmetric convolution layer corresponding to size $1 \times w$ with filter $W_k^2 \in \mathbb{R}^{1 \times w \times m}$ is used to convolve w successive slices as times step centered at position i that are denoted as F_i, that is $F_i = (f_{i-m/2}^1, \cdots, f_{i-1}^1, f_i^1, f_{i+1}^1, \cdots, f_{i+m/2}^1) \in \mathbb{R}^{1 \times w \times m}$ where $f_i^1 = f^1(:, i, :)$ or zero tensor with the same shape if $i \notin \{1, \cdots, l\}$, then a local feature is produced as below:

$$f_{ik}^2 = \max(W_k^2 * F_i + b, 0) \tag{9}$$

where subscript $k \in \{1, \cdots, m\}$ is used to index the channels of the feature map and superscript 2 denoted the second convolution, others are the same as described above. So the feature maps in the second phase are a tensor $f^2 \in \mathbb{R}^{1\times l\times m}$.

In the BLSTM-Inception module which is a block shown in Fig.1 with red dash line as boundary, three asymmetric convolution layers are used simultaneously to the output tensor H produced by BLSTM with the width of convolution filter $w \in \{3, 5, 7\}$ which is proved to achieve great results in the experiments and in coincidence with the simple CNN architecture proposed by Kim, then three tensors with the same shape as $f^2 \in \mathbb{R}^{1\times l\times m}$ corresponding to $w \in \{3, 5, 7\}$ are concatenated in channels as a tensor $f^* \in \mathbb{R}^{1\times l\times 3m}$. Actually, we find that if squeeze the first dimensionality in f^*, it can be token as input to BLSTM again. So, the BLSTM-Inception module can be stacked to get more complex architecture as Inception v3 [18].

C. Global Max Pooling Layer

Then global max pooling operation rather than global average pooling which is used in NIN [8] and GoogLeNet [17] is applied to generate the representation of a sentence or document:

$$h^* = down_{global}(f^*) \quad (10)$$

where $down_{global}(\cdot)$ stands for global max pooling , that generate a feature from each feature map. So h^* is a fixed-length vector representation whose dimensionality is $3m$.

D. Output Layer

For text classification, the output h^* of global max pooling layer is passed to the softmax layer to get a predicted probability distribution over the discrete set of class labels:

$$p = softmax(Wh^* + b) \quad (11)$$

Given the training set$\{(D^{(i)}, y^{(i)})|i = 1, \cdots, N\}$, where $D^{(i)}$ denotes the t-th document and $y^{(i)} \in \{1, \cdots, \mathrm{K}\}$ (K is the number of possible target labels), N denotes the number of training samples, then the cost function used to train the proposed neural network architecture is the categorical cross-entropy loss with $L2$ constraint on the weight matrix in the softmax layer:

$$J(\theta) = -\frac{1}{N}\sum_{i=1}^{N} \log p_{y^{(i)}}^{(i)} + \lambda \|W\|_F{}^2 \quad (12)$$

where θ denotes all parameters in the network that will be learnt, $p_{y^{(i)}}^{(i)}$ is the $y^{(i)}$ -th element of the predicted distribution for $D^{(i)}$, λ is the penalty coefficient.

E. Regularization

To prevent over-fitting, dropout is applied on the word embeddings that will corrupt the captured semantic and syntactic relation in the embedding space to learn better and robust representations of words for text classification, on the double channels produced by BLSTM and penultimate layer that will prevent co-adaptation of hidden units. Besides that, the $L2$-norm of weight matrix in the softmax layer is constrained by adding $\|W\|_F{}^2$ as a penalty term to the cost function.

III. EXPERIMENTAL SETUP

A. Datasets

The proposed model BLSTM-Inception v1 is evaluated on several benchmarks, and the summary statistics of the datasets are listed in Table 1.

- MR: Movie reviews with one sentence per review. Classification involves detecting positive and negative reviews [14].
- SST-1: Stanford Sentiment Treebank-an extension of MR but with train/dev/test splits provided and fine-grained labels (very positive, positive , neutral, negative, very negative) relabeled by Socher et al. [15].
- SST-2: Same as SST-1 but with neutral reviews removed and binary labels. For both experiments, extra phrase data with sentiment labels that comes from [5] is only added to training set.
- Subj: Subjectivity dataset where the task is to classify a sentence as being subjective or objective [13].
- TREC: TREC question dataset-task involves classifying a question into 6 question types (where the question is about person, location, numeric information, etc) [6].

B. Word Embeddings

Initialize the embedding layer with word vectors pre-trained on much larger unannotated corpora is a popular way to improve the generalization ability in the absence of a large supervised training set. In the experiments, the publicly available word2vec vectors [11] trained on 100

Table 1: Summary of statics for the datasets.

Data	c	l	m	train	dev	test	$\|V\|$	$\|V_{pre}\|$
SST-1	5	18	51	8544	1101	2210	17836	12745
SST-2	2	19	51	6920	872	1821	16185	11490
MR	2	21	59	10662	-	**CV**	20191	16746
Subj	2	23	65	10000	-	**CV**	21057	17671
TREC	6	10	33	5452	-	500	9137	5990

c: Number of target classes. l: Average sentence length. m: Maximum sentence length. train/dev/test: size of train/development set, CV in test column means 10-fold cross validation. $|V|$: Vocabulary size. $|V_{pre}|$: Number of words in the set of pre-trained word vectors.

billion words from Google News are used. The dimensionality of embedding space is 300, and words not in the set of pre-trained words are initialized by randomly sampling from uniform distribution in $[-0.25, 0.25]$.

C. Hyper-parameters and Training Details

The dimensionality of hidden state of BLSTM is 300 which equals to the dimensionality of the word embeddings. 100 convolutional filters are employed in both of two convolution operations in the asymmetric convolution layer corresponding to width $w \in \{3, 5, 7\}$. As for regularization, dropout is applied with dropout rate of 0.5 for word embeddings, 0.2 for the output of BLSTM, 0.4 for the penultimate layer, and the penalty coefficient 10^{-6} is used. These hyper-parameters are chosen via a grid search on the SST-2 development set.

For dataset without a standard development set, 10% of the training data is randomly selected. The stochastic gradient descent with the Adadelta update rule [21] is used over shuffled mini-batches to train the model, and the initial learning rate is set as 0.1. The neural network architecture is implemented on the basis of tensorflow library.

IV. Results and Analysis

A. Overall Performance

The performances of BLSTM-Inception v1 and other state-of-the-art models on five tasks including sentiment classification, question classification and subjectivity classification are listed in Table 2. The BLSTM-Inception v1 achieves state-of-the-art results on 4 out of 5 tasks. It not only performs best on both of SST-1 and SST-2, but also improves the result on the MR by nearly 1.5% compared to BLSTM-2DCNN. The combination of BLSTM and asymmetric factorization of normal convolution obviously contribute to the excellent results. Besides that, as shown in following comparative experiments, the extra phrase data in SST-1 and SST-2, the way to use output of BLSTM layer and global max pooling are also crucial.

B. Effect of extra phrase data and Batch Normalization in SST

In recent years, some models have used the extra phrase data in both of SST-1 and SST-2 to improve the performance, but not explore the effect of it.

Besides that, BLSTM-Inception v1 (BN) is implemented where batch normalization [2] is used in every convolution before nonlinearity in BLSTM-Inception v1. The results are listed in Table 3. Even if extra phrase data is not used, BLSTM-Inception v1 can still achieve great results comparable to state-of-the-art. The extra phrase data generally improve more than 1%. It's interesting that batch normalization combined with phrase data can improve more, but worse when phrase data is not used. Actually, batch normalization will introduce extra parameters, so more data needs to reflect its ability of improving generalization.

C. EquationsSingle Channel vs. Double Channels

The hidden state matrices from forward and backward LSTM are added as single channel in BLSTM-2DCNN, but are concatenated as double channels in BLSTM-Inception v1. The experiment results listed in Table 5 show that the proposed double channels mode outperforms single channel mode. Actually, the single channel mode could lead to the loss of information that may directly reduce the performance of asymmetric convolutional layer.

Table 2: Classification Results on five standard benchmarks.

Model	SST-1	SST-2	MR	Subj	TREC
CNN-non-static [4]	48.0	87.2	81.5	93.4	93.6
CNN-multichannel [4]	47.4	88.1	81.1	93.2	92.2
Molding-CNN [5]	51.2	88.6	-	-	-
TBCNN [12]	51.4	87.9	-	-	-
C-LSTM [24]	49.2	87.8	-	-	94.6
Tree-LSTM [19]	51.0	88.0	-	-	-
MVCNN [20]	49.6	89.4	-	93.9	-
Multi-Task [9]	49.6	87.9	-	94.1	-
DSCNN [23]	49.7	89.1	81.5	93.2	95.4
BLSTM-2DCNN [25]	52.4	89.5	82.3	94.0	**96.1**
BLSTM-Inception v1	**52.62**	**90.12**	**83.86**	**94.49**	95.40

CNN-non-static/CNN-multichannel: Convolutional neural network for sentence classification [4]. Molding-CNN: Molding CNNs for text: non-linear, non-consecutive convolution [5]. TBCNN: Discriminative neural sentence modeling by tree-based convolution [12]. C-LSTM: A C-LSTM Neural Network for text classification [24]. Tree-LSTM: Improved semantic representations from tree-structured long short-term memory networks [19]. MVCNN: Multichannel variable-size convolution for sentence classification [20]. Multi-Task: Recurrent Neural Network for Text Classification with Multi-Task Learning [9]. DSCNN: Dependency Sensitive Convolutional Neural Networks for Modeling Sentences and Documents [23]. BLSTM-2DCNN: Text Classification Improved by Integrating Bidirectional LSTM with Two-dimensional Max Pooling [25].

Table 3: Accuracy of BLSTM-Inception v1 and variant on SST

Model	SST-1	SST-1(phrase)	SST-2	SST-2(phrase)
BLSTM-Inception v1	51.36	52.62	89.02	90.12
BLSTM-Inception v1(BN)	50.32	52.81	88.41	90.61

Table 4: Global max pooling compared to flatten and local max pooling

Method	SST-1	SST-2
Flatten	47.60	86.66
Local Max Pooling (1x7)	49.77	87.75
Global Max Pooling	51.36	89.02

Table 5: Double channels compared to single channel

Mode	SST-1	SST-2
Single Channel	50.63	88.63
Double Channels	51.36	89.02

D. Global Max Pooling as a Regular

Finally, instead of flattening the output of BLSTM-Inception module, the global max pooling is employed. Besides that, local max pooling with window size 1x7 is also tried. The experiment results listed in Table 4 and shown in Fig.2 demonstrate that global max pooling compared to flatten and local max pooling (1x7) achieves better generalization. The global max pooling extracts the most important feature from each feature map which avoids the effect of other noise features and greatly decreases the number of parameters in the softmax layer that are effective to prevent over-fitting.

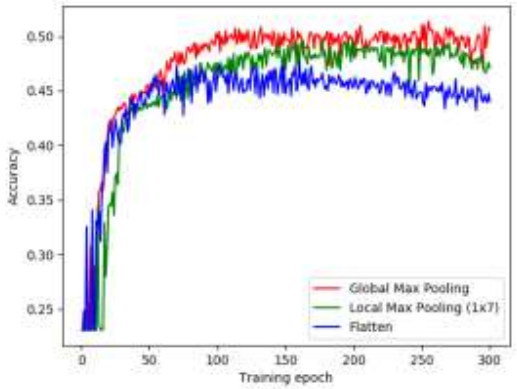

Figure 2: Accuracy on SST-1 test set in the first 300 epochs with above three methods is shown.

V. CONCLUSIONS

In this paper, a novel neural network architecture named BLSTM-Inception v1 is proposed. Its core is the BLSTM-Inception module which integrates BLSTM with multiple asymmetric convolutional layer of different scales similar to Inception modules [18]. Before efficiently and excellently extracting the features of n-gram phrases with different width, this module has incorporated the past and future information into the word representation by BLSTM. Besides that, a series of comparative experiments show that the use of extra phrase data, the proposed double channels mode and global max pooling are also effective methods to improve performance. Despite little tuning of hyper-parameters, BLSTM-Inception v1 architecture achieves new state-of-the-art on 4 of 5 tasks, and especially improves nearly 1.5% on MR.

ACKNOWLEDGMENT

We want to thank for the suggestions from some anonymous reviewers. This work is partially supported by National Natural Science Foundation of China under Grant Nos. 61373063, 61375007, 61233011, 91420201, 61472187 and by National Basic Research Program of China under Grant No.2014CB349303.

REFERENCES

[1] Sepp Hochreiter and J¨urgen Schmidhuber. “Long short-term memory,” Neural Computation, 9(8): 1735–1780, 1997.

[2] Sergey Ioffe and Christian Szegedy. “Batch normalization: accelerating deep network training by reducing internal covariate shift,” In ICML, pages 448–456, 2015.

[3] Nal Kalchbrenner, Edward Grefenstette and Phil Blunsom. “A convolutional neural network for modelling sentences,” In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers, pages 655–665, 2014.

[4] Yoon Kim. “Convolutional neural networks for sentence classification,” In EMNLP, pages 1746–1751, 2014.

[5] Tao Lei, Regina Barzilay and Tommi Jaakkola. “Molding cnns for text: non-linear, non-consecutive convolutions,” In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, September 2015.

[6] Xin Li and Dan Roth. 2002. “Learning question classifiers,” In Proceedings of the 19th international conference on Computational linguistics-Volume 1, pages 1–7. Association for Computational Linguistics.

[7] Depeng Liang and Yongdong Zhang. “AC-BLSTM: asymmetric convolutional bidirectional LSTM networks for text classification,” In arXiv preprint arXiv: 1611.01884, 2016.

[8] Min Lin, Qiang Chen and Shuicheng Yan. “Network in network,” CoRR, abs/1312.4400, 2013

[9] Pengfei Liu, Xipeng Qiu and Xuanjing Huang. “Recurrent neural network for text classification with multi-task learning,” In Proceedings of International Joint Conference on Artificial Intelligence (IJCAI), 2016.

[10] Marcus Liwicki, Alex Graves, Horst Bunke and J¨urgen Schmidhuber. “A novel approach to online handwriting recognition based on bidirectional long short-term memory networks,” In Proc. 9th Int. Conf. on Document Analysis and Recognition, volume 1, pages 367–371, 2007.

[11] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado and Jeff Dean. “Distributed representations of words and phrases and their compositionality,” In Proceedings of NIPS 2013.

[12] Lili Mou, Hao Peng, Ge Li, Yan Xu, Lu Zhang and Zhi Jin. “Discriminative neural sentence modeling by tree-based convolution,” In EMNLP, pages 2315–2325, 2015.

[13] Bo Pang and Lillian Lee. 2004. “A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts,” In Proceedings of the 42nd annual meeting on Association for Computational Linguistics, page 271. Association for Computational Linguistics.

[14] Bo Pang and Lillian Lee. 2005. “Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales,” In Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, pages 115–124. Association for Computational Linguistics.

[15] Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng and Christopher Potts. 2013. "Recursive deep models for semantic compositionality over a sentiment Treebank," In EMNLP, volume 1631, page 1642. Citeseer

[16] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever and Ruslan Salakhutdinov. "Dropout: a simple way to prevent neural networks from overfitting," Journal of Machine Learning Research, 15(1):1929–1958, 2014.

[17] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke and Andrew Rabinovich. "Going deeper with convolutions," In CVPR 2015, Boston, MA, USA, June 7-12, 2015, pages 1–9, 2015a.

[18] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens and Zbigniew Wojna. "Rethinking the inception architecture for computer vision," CoRR, abs/1512.00567, 2015b

[19] Kai Sheng Tai, Richard Socher and Christopher D Manning. "Improved semantic representations from tree-structured long short-term memory networks," In Proceedings of NIPS, pages 1556-1566, 2015.

[20] Wenpeng Yin and Hinrich Sch¨utze. "Multichannel variable-size convolution for sentence classification," In CoNLL, pages 204-214, 2015.

[21] Matthew D Zeiler. "Adadelta: An adaptive learning rate method," In CoRR, 2012.

[22] Wojciech Zaremba, Ilya Sutskever and Oriol Vinyals. "Recurrent neural network regularization," In arXiv preprint arXiv: 1409.2329, 2014.

[23] Rui Zhang, Honglak Lee and Dragomir Radev. "Dependency sensitive convolutional neural networks for modeling sentences and documents," In Proceedings of NAACL-HLT, pages 1512–1521, 2016.

[24] Chunting Zhou, Chonglin Sun, Zhiyuan Liu and Francis C. M. Lau. "A C-LSTM neural network for text classification," CoRR, abs/1511.08630, 2015.

[25] Peng Zhou, Zhenyu Qi, Suncong Zheng, Jiaming Xu, Hongyun Bao and Bo Xu. "Text classification improved by integrating bidirectional LSTM with two-dimensional max pooling," In COLING, 2016.