2017 International Conference on Security, Pattern Analysis,
and Cybernetics (SPAC)

A Convolutional Attentional Neural Network for Sentiment Classification

Jiachen Du[1,2], Lin Gui[1], Yulan He[3], Ruifeng Xu[1](✉)
[1] School of Computer Science and Technology
Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen, China
[2] Department of Computing, the Hong Kong Polytechnic University, Hong Kong
[3] School of Engineering and Applied Science, Aston University, United Kingdom
dujiachen@stmail.hitsz.edu.cn, guilin.nlp@gmail.com, y.he9@aston.ac.uk, xuruifeng@hit.edu.cn

***Abstract*—Neural network models with attention mechanism have shown their efficiencies on various tasks. However, there is little research work on attention mechanism for text classification and existing attention model for text classification lacks of cognitive intuition and mathematical explanation. In this paper, we propose a new architecture of neural network based on the attention model for text classification. In particular, we show that the convolutional neural network (CNN) is a reasonable model for extracting attentions from text sequences in mathematics. We then propose a novel attention model base on CNN and introduce a new network architecture which combines recurrent neural network with our CNN-based attention model. Experimental results on five datasets show that our proposed models can accurately capture the salient parts of sentences to improve the performance of text classification.**

I. Introduction

In recent years, there is no doubt that deep learning has ushered in amazing technological advances on natural language processing(NLP) researches. Much of the work with deep learning involves learning word vector representations through natural language models [1], [2], [3] and composition over word vectors for various tasks like text classification [4], machine translation [5], document summarization [6] and so on.

In this paper, we focus on the text classification problem. Traditional approaches to text classification firstly represent text sequences with sparse features, such as n-grams, topic-based representation [7] and kernel methods [8]. Recently deep-learning models have shown their big success in text classification, such as convolutional neural networks [9] and recurrent neural networks based on long short-term memory [10].

Applying convolutional neural network (CNN) to NLP including text classification has drawn many interests in recent years. It has been shown that CNNs can be directly applied to embeddings of words [11], [12], [13]. Unlike word level models, [14] proposed a character-level CNN for text classification which achieved competitive results. Although CNN has been proven efficient on text classification, it usually ignores important long-distance sequential information which greatly impacts the classification performance, especially in sentences which have negation and semantic transition. Recurrent neural network (RNN) is another important model in NLP. [15] used gated recurrent neural network to model documents and applied this model to sentiment classification. [16] explored the structure of a sentence and used a tree-structured recurrent neural network with long-short term memory (LSTM) for text classification. The advantage of RNN is its ability to better capture the contextual information, especially the semantics of long texts. However RNN model cannot pay attention to the salient parts of text. This limitation reduces the effectiveness of RNN when applied to text classification.

Recently, based on the aforementioned architectures, a new direction of neural networks has emerged. It learns to focus "attention" to specific parts of text as the simulation of human's attention while reading. However, the researches on neural network with attention mechanism only show promising results on a sequence-to-sequence (seq2seq) tasks in NLP, including machine translation [17], caption generation [18] and text summarization [19]. It's not appropriate to use the same alignment mechanism for classification. For example, [20] applies the attention model used in seq2seq tasks to document-level classification. The attention is modeled by a one hidden layer perceptron with RNN hidden unit as input. Compare with non-attention methods, the improvement is limited in the reported result. The reason is that, in the seq2seq problem, each word has a corresponding label, such as the word in another language (for machine translation tasks), or a Part-Of-Speech label (for POS tagging tasks). In text classification, there are no target labels for each word to indicate the word is category-relevant or not. So, there is no evidence showing why the perceptron with one hidden layer attention is efficient for text classification.

Motivated by the cognitive and neuroscience research, we propose a novel attention model. The basic idea is to first use the convolution operation to capture attention signals, each of which stands for the local information of a word in its context; then use RNN to model text with attention signals. A word with higher attention weight, which carries more valuable information, will be more important in text modeling. Our main contributions in this paper is three-fold:

- This is the first time a convolutional neural network model is presented to stimulate human's reading attention based on cognitive and neuroscience research, and a detailed mathematical deduction of this model is given

978-1-5386-3016-7/17/$31.00 ©2017 IEEE

in this work.

- Moreover, a novel attention extraction method based on this model has been proposed that can be used in several tasks.
- Finally, we propose a new architecture based on convolutional attention extraction model, and this neural network shows competitive results compared with state-of-the-art models in text classification.

The rest of our paper is structured as follows, Section 2 explores a novel attention model based on convolutional neural networks and gives detailed reduction of this model. Section 3 gives a description of the Convolutional Recurrent Attention Network (CRAN) for text classification. Section 4 presents some experiments to justify the effectiveness of CRAN on text classification. Section 5 concludes the paper and outlines the future work.

II. A CNN-BASED ATTENTION MODEL

It is commonly known that we can pay attention to only small amount of information presented in visual scenes and only concentrate on the information related to a specific task at hand. Cognitive and neuroscience researches have confirmed this hypothesis, and a lot of experiments have shown that humans depict in brains a visual representation or "search template" of certain task and try to only pay attention to the information which can match the "task-oriented template" [21]. psycholinguistics has proven that template-matching process also helps us emphasize the important content while our brain is processing texts. Although this mechanism of attention has been thoroughly studied in neuroscience and psychology, there is few research on how to introduce it to computational linguistic and natural language processing. Motivated by that, we propose a novel model introducing the aforementioned attention mechanism to natural language processing , especially to text classification.

Based on in-depth investigation, we found that Convolutional Neural Network (CNN) is a natural model to stimulate human being's reading attention mechanism , since the convolution operation as the core component of CNN is similar to the process of template matching. For textual data, CNN always applies one-dimensional convolution to the concatenate of vector representation of each words. Our first goal is to show that one-dimension convolution of CNN is precisely the process of calculating the similarity between snippets of text and the "attention searching templates". In neural-network based models, a text sequence of length T (padded where necessary) is often represented as

$$x_{0:T-1} = x_0 \oplus x_1 \oplus \ldots \oplus x_{T-1} \tag{1}$$

where $x_t \in \mathcal{R}^d$ $(t = \{0, 1, \ldots, T-1\})$ corresponds to the d-dimensional vector representation of the t-th word in the text sequence, and $\oplus$ is the vector concatenation operator. Convolution operation applies a filter $w = w_0 \oplus w_1 \oplus \ldots \oplus w_{l-1}$ to a window of l words in the original sequence to get a convolutional similarity c_t, where each column of the filter $w_t \in \mathcal{R}^d$ $(t = \{0, 1, \ldots, l-1\})$ is a vector of the same length of word embeddings. The convolution operation apply the following transformation to each continuous subsequence of length l in $[x_0, x_1, \ldots, x_{T-1}]$. Suppose the current subsequence is $x_{t:t+l-1} = x_t \oplus x_{t+1} \oplus \ldots \oplus x_{t+l-1}$:

$$\begin{aligned} c_t &= f\left(< w, x_{t:t+l-1} > + b\right) \\ &= f\left(||w|| \times ||x_{t:t+l-1}|| \times \frac{< w, x_{t:t+l-1} >}{||w|| \times ||x_{t:t+l-1}||} + b\right) \end{aligned} \tag{2}$$

In equation 2, $< \cdot, \cdot >$ is dot product of two vectors as $< a, b >= a^T b$, $|| \cdot ||$ is the $\mathcal{L}^2$ norm of vectors. f is the non-linear function such as hyperbolic tangent, sigmoid and rectified linear function. Notice that w and $x_{t:t+l-1}$ are $l \times d$-dimensional vector, assuming that each dimension has its own distribution. According to Chebyshev Law, there exists M, for any w and any $x_{t:t+l-1}$, if $l \times d$ is large enough (usually larger than 25), it is true that for any $\varepsilon > 0$, $P(|M - ||w|| \times ||x_{t:t+l-1}||| < \varepsilon) = 1$. since $l \times d$, namely the shape of convolution filters is always larger than 25, we can replace $||w|| \times ||x_{t:t+l-1}||$ by M in this equation, and define a function $F(x) = f(Mx)$ to replace the original function f, let $b' = b/M$, we will obtain:

$$c_t = F(\cos(w, x_{t:t+l-1}) + b') \tag{3}$$

In equation 3 we notice that F is only a compression function that satisfies $F'(x) > 0$. Then we can consider the convolutional filter w be the "search template" mentioned before in human's attention while reading. And c_t can be treated as the cosine similarity between the search template and the part of text which is processed currently. b' in equation 3 is the threshold of this similarity. When the similarity is greater than b', the textual part being processed is considered task-relevant; otherwise it is task-irrelevant.

The output of each convolutional filter is an attention signal from the original text. In order to reduce the noise in the attention signals, we choose multiple convolutional filters applied to the vector representation of original text and average the results of these filters to obtain a smooth attention signal. Suppose the number of convolutional filters is m, these filter is denoted by $[w^1, w^2, \ldots, w^m]$, and the corresponding attention similarity is $[c^1, c^2, \ldots, c^m]$. After averaging the attention similarities along the filter-axis, we will obtain the smooth attention signal $c \in \mathcal{R}^T$ of which each element represents the importance of the corresponding word.

$$c = \sum_{i=1}^{m} c^i \tag{4}$$

The whole CNN-based attention model is shown in Figure 1.

III. CONVOLUTIONAL-RECURRENT ATTENTION NEURAL NETWORKS

Based on the attention model we described in Section 2, we propose a model named Convolutional-Recurrent Attention Network (CRAN) that combines recurrent neural network (RNN) and the convolutional attention model in Section

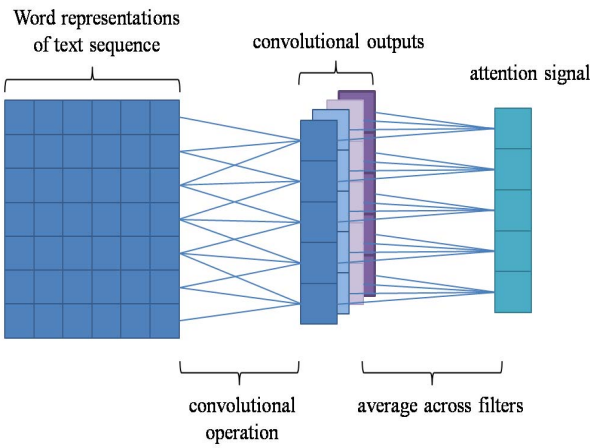

Fig. 1. CNN-based model of attention extraction.

2. The reason why we use RNN as a part of our model rather than directly using the traditional CNN architecture is that traditional CNN uses a pooling layer over the whole sentence which results in a single vector being extracted as the representation of the sequence. It makes CNN difficult to capture the long-distance dependencies in sequences like negation and transition. However RNN with long short-term memory (LSTM) is designed for handling the long-distance dependency problem. We speculate that combining RNN with our proposed CNN-based attention model will give better performance compared with using a pure CNN model, and the experimental results which will be presented in Section 4 confirm our hypothesis.

The overall architecture of the Convolutional Attention Neural Network (CRAN) is shown in Figure 2. It consists of two main parts: a recurrent neural network (RNN) as the text encoder and a convolutional neural network (CNN) as the attention extractor. We describe the details of these two parts in the following subsections.

A. RNN-Based Sequence Encoder

An RNN [22] is a kind of neural network that processes sequences of arbitrary length by recursively applying a function to its hidden state vector $h_t \in \mathcal{R}^d$ of each element in the input sequences. The hidden state vector at time-step depends on the input symbol x_t and the hidden state vector at last time-step h_{t-1} is:

$$h_t = \begin{cases} 0 & t = 0 \\ g(g_{t-1}, x_t) & \text{otherwise} \end{cases} \tag{5}$$

A fundamental problem in traditional RNN is that gradients propagated over many steps tend to either vanish or explode. It makes RNN difficult to learn long-dependency correlations in a sequence. Long short-term memory network (LSTM) was proposed by [10] to alleviate this problem. LSTM has three gates: an input gate i_t , a forget gate f_t , an output gate o_t and a memory cell c_t. They are all vectors in $\mathcal{R}^d$. The LSTM transition equations are:

$$\begin{aligned} i_t &= \sigma(W_i x_t + U_i h_{t-1} + V_i c_{t-1}), \\ f_t &= \sigma(W_f x_t + U_f h_{t-1} + V_f c_{t-1}), \\ o_t &= \sigma(W_o x_t + U_o h_{t-1} + V_o c_{t-1}), \\ \tilde{c}_t &= \tanh(W_c x_t + U_c h_{t-1}), \\ c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t, \\ h_t &= o_t \odot \tanh(c_t) \end{aligned} \tag{6}$$

where x_t is the input at the current time step, σ is the sigmoid function and $\odot$ is the elementwise multiplication operation. In our model, we use the hidden-state vector of each time step as the representation of corresponding word in the sentence.

B. CNN-Based Attention Extraction

As discussed in Section 2, we use a CNN-based network to model the attention signal in sentences. Suppose the input text sequence is $[x_0, x_1, \ldots, x_{T-1}]$, where $x_t \in \mathcal{R}^d$ $(t = 0, 1, \ldots, T-1)$, m convolutional filters of length l are denoted as $[w^1, w^2, \ldots, w^m]$, the corresponding convolution results are $[c^1, c^2, \ldots, c^m]$, where $c^i \in \mathcal{R}^T$ $(t = 1, 2, \ldots, m)$ represents the attention distributed on the sequence of length T. We average the m filters to get the stable attention signal c witch is a vector of length T as described in Section 2.

C. Text Classification

We use the product of attention signal c_t and the corresponding hidden state vector of RNN h_t to represent the word t in a sequence with attention signal. The representation of the whole sequence can be obtained by averaging the word representations:

$$s = \frac{1}{T} \sum_{t=0}^{T-1} c_t h_t \tag{7}$$

where $s \in \mathcal{R}^d$ is the vector representation of the text sequence and it can be used as features for text classification:

$$p = \text{softmax}(W_c s + b_c) \tag{8}$$

where p is the predicted label for text sequence s , W_c and b_c are parameters of the classification layer.

IV. Experiments

In this section, we investigate the empirical performance of our proposed CRAN on various datasets and compare it with state-of-the-art models for text classification.

A. Datasets

We choose five different text classification datasets and test our model on various benchmarks. We briefly summarize these datasets as follows:

- **MR**: Movie reviews with one sentence per review. Classification involves binary categories of reviews (positive and negative) [1] [23].

[1] http://www.cs.cornell.edu/people/pabo/movie-review-data/

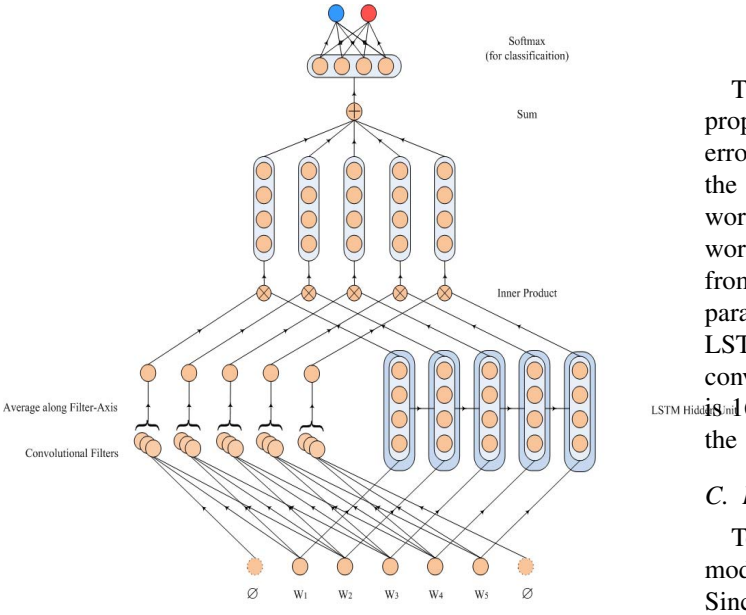

Fig. 2. Architecture of convolutional attention network

Dataset	Class	Avg Length	Vocabulary	Text Size	Test Size
MR	2	20	18K	10662	CV
SST-1	5	19	18K	11855	2210
SST-2	2	18	15K	9613	1821
Subj	2	21	21K	10000	CV
IMDB	2	294	392K	50000	25000

TABLE I
SUMMARY STATISTICS OF THE FIVE DATASETS.

- **SST-1**: An extension of MR but with train/dev/test splits provided and fine-grinned labels (negative, somewhat negative, neutral, somewhat positive, positive) [2] [24].
- **SST-2**: Same as SST-1 but with neural reviews removed and only containing binary labels (positive and negative).
- **Subj**: Subjectivity dataset where task is to classify a sentence as objective or subjective [25].
- **IMDB**: A document-level text classification dataset containing 100,000 movie reviews with binary labels [3] [26].

The first four datasets are for sentence-level classification and the last one is for document-level classification. We conduct experiments on the standard test sets on SST-1,SST-2 and IMDB. For datasets without standard train/test split, we use 10-fold cross validation instead. The summary statistics of the datasets are listed in Table 1.

B. Model Training and Hyper-parameters

We train our proposed convolutional attention model with two different initialization strategies:

- **CRAN_rand** : The convolutional attention layer is randomly initialized.
- **CRAN_pretrain** : The convolutional attention layer is pre-trained by a standard standard convolutional neural network which is connected to a fully-connected layer to the final labels as a classifier.

The model can be trained in an end-to-end way by back-propagation, where the objective function is cross-entropy of error loss. Training is done through gradient descent with the Adadelta update rule. In all of these experiments, the word embeddings are initialized with the publicly available word2vec vectors that were trained on 100 billion words from Google News (Mikolov, Sutskever, et al., 2013) . Other parameters are set as follows. The number of hidden units of LSTM and convolution filters in CNN is 100 , the length of convolution filter is 3, dropout rate is 0.5 and mini-batch size is 16. These hyper-parameters are chosen via a grid search on the SST-2 development set.

C. Baselines

To illustrate the performance boost of our proposed attention model, we compare our model with some baseline methods. Since we use RNN as component of our model, we implement an RNN with LSTM memory unit as a baseline. We also compare our model with LSTM with the attention model proposed by [20].

- **LSTM**: LSTM for text classification in [27].
- **LSTM + RNN attention** attention: The attention-based LSTM model proposed by [20]. Since most datasets used in our experiments are for sentence-level classification, we implement a flatten variant of this model without aggregating the attention signals from sentences to form the document-level attention signal.

We also compare our model with the following state-of-the-art models:

- **NBOW**: The NBOW sums the word vectors within the sentence and applies a softmax classifier.
- **MV-RNN**: Matrix-Vector Recursive Neural Network with parse trees [28].
- **F-Dropout**: Fast Dropout from [29].
- **DCNN**: Convolutional Neural Network with dynamic k-max pooling [30].
- **CNN**: Convolutional neural networks with max pooling [13].
- **Tree-LSTM**: Tree-Structured Long Short-Term Memory Networks [16].
- **Multi-Task**: Shared-layer multi-task learning model trained on four different datasets [31].

D. Results and Analysis

The experimental results on all datasets are shown in Table 2. Firstly we notice that LSTM performs the worst among all the models since the sequential features extracted by RNN models is not suitable for text classification. But we find that our proposed CRAN can combine the merits of CNN and RNN to improve the performance. Results show that CRAN improves the performance significantly compared with the LSTM by 3.0% on average. For the two variations of CRAN, we can see that CRAN_rand model performs slightly better

[2]http://ai.stanford.edu/sentiment/
[3]http://ai.stanford.edu/ amaas/data/sentiment/

Model	MR	SST-1	SST-2	Subj	IMDB
CRAN_rand	**82.8**	50.0	87.7	**94.1**	92.0
CRAN_pretrain	82.0	48.1	86.9	94.0	**92.1**
LSTM	80.1	46.2	85.2	91.2	88.5
LSTM + RNN attention	82.0	48.0	86.1	93.2	90.6
NBOW	77.1	42.1	79.0	90.8	80.7
MV-RNN	79.0	44.4	82.9	-	-
F-Dropout	79.1	-	-	93.6	91.1
DCNN	-	48.5	86.8		
CNN	81.5	48.0	**88.1**	93.4	-
Tree-LSTM*	-	**50.6**	86.9	-	-
Multi-Task*	-	49.6	87.9	**94.1**	91.3

TABLE II
RESULTS OF OUR PROPOSED CRAN AGAINST BASELINES. RESULTS MARKED * ARE MODELS THAT NEED EXTERNAL TOOLS OR RESOURCES.

than CRAN_pretrain model except on the IMDB dataset. This is because CRAN_pretrain uses a different architecture to pre-train the convolutional attention layer. Both CRAN variants perform better than the LSTM with RNN attention. This shows that our proposed attention modelling method can capture more accurate attention information from text.

Compared with the state-of-the-art models in text classification, CRAN gives the best performance on three out of five datasets. Although CRAN uses a similar CNN network as its attention extractor, CRAN improves upon CNN by 0.9% on average. This is because combining RNN helps our model capture the long-distance semantic dependencies in sentences which cannot be dealt with by traditional CNNs. Tree-LSTM outperforms our model on SST-1 by 0.6%. However it needs an external parser to derive the tree-structure of each sentence, and the performance listed in Table 2 is reported on the exact parsing results of sentences labelled by annotators. It is worth noting that our models are comparable with the recurrent neural network with multi-task learning proposed by (Liu et al., 2016). This model is an extremely strong baseline which was trained jointly on four datasets (SST-1, SST-2, Subj and IMDB).

E. Case Study

In order to validate that our model is able to select salient parts in a text sequence, we visualize the attention layers in Figure 3 for several sentences from the MR dataset in which our model predicted the class labels correctly. Each line in the figure is a heatmap of the attention signals extracted by our model. The attention signal is normalized in the range of [-1,1], and the actual label of each sentence is shown to the left of each heatmap. The red color in the heatmap means a high attention for the positive label and the blue color means a high attention for the negative label, while the white color means neural.

Figure 3 shows that our model can not only select words carrying strong sentiments like *repelling, annoyance, dull, et.al.*, but also deal with transitions of sentiments in sentences. For example, in the second sentence, our model assigns a high positive attention to the first half of the sentence before the word *but*, and it also finds the second half of this sentence is highly related to the negative sentiment. Also, the negative attention value is greater than the positive attention value in this sentence, which makes the final predicted label to be negative.

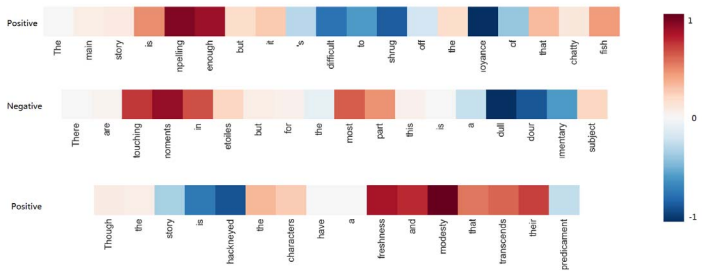

Fig. 3. Visualization of attention signals in sample sentences in the MR dataset.

We also analyze the sentences from which our model failed to extract attention signals properly. We show an attention visualization in Figure 4 which is extracted from a sentence in the MR dataset:

Imagine the James Woods character from video drome making a home movie of audrey rose and showing it to the kid from the sixth sense and youve imagined the ring.

The label of this sentence is negative, but our model predicts its labels positive. We can see that this is a complex sentence with some metaphors. To understand the real meaning of this sentence, readers must have the background knowledge of the movies mentioned in this sentence. For people who are unfamiliar with the movies, James Woods and Sixth Sense, they will not know these are all horror movies and hence would have a difficulty in understanding the metaphoric meaning of these terms.

We can obverse that, for this sentence, the attention signals extracted by our model is somewhat randomized and do not reflect the actual sentiment expressed in the sentence. It seems that our model can only extract the attention signals from the literal meanings of words. Nevertheless, understanding complicated linguistic phenomena such as analogy, metaphor and irony is a huge challenge commonly faced by many NLP tasks, not just in text classification.

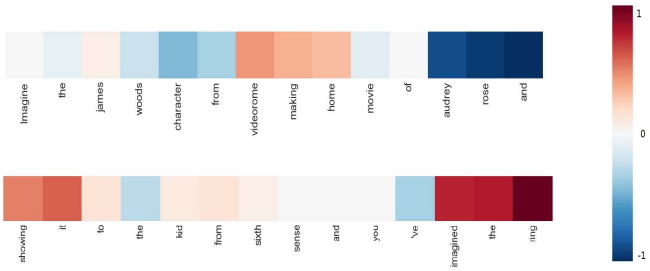

Fig. 4. Visualization of wrongly classified sentence in the MR dataset.

V. Conclusion

In this paper, we have shown that the convolutional neural network is a reasonable model for extracting attentions from text sequences. Based on this finding, we have proposed a novel attention extraction model based on the convolutional neural network. Utilizing this CNN-based attention model, we have introduced a new neural network architecture which combines RNN with our CNN-based attention model. We have conducted extensive experiments on five datasets. The experimental results show that (1) our model is capable of extracting salient parts from sentences; (2) our model can combine the merits of CNN and RNN to improve the sentence classification performance. Finally, the visualization of some attentions extracted by our model shows the impressive capability of our model to process the sentiment transitions in sentences. We have also presented an attentional visualization result of a sentence whose class label was wrongly predicted by our model. This shows that our model has a difficulty in dealing with more subtle meanings embedded in sentences, a huge challenge commonly faced by many NLP tasks. In future works, we will mainly focus on extending our CNN-based attention model to other tasks like text generation and sequence to sequence learning.

Acknowledgements

This work was supported by the National Natural Science Foundation of China 61370165, U1636103, 61632011, Shenzhen Foundational Research Funding JCYJ20150625142543470, 20170307150024907, Guangdong Provincial Engineering Technology Research Center for Data Science 2016KF09.

References

[1] Y. Bengio, R. Ducharme, and P. Vincent, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, 2003.

[2] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[3] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, Conference Proceedings, pp. 3111–3119.

[4] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of Machine Learning Research*, vol. 12, no. Aug, pp. 2493–2537, 2011.

[5] N. Kalchbrenner and P. Blunsom, "Recurrent continuous translation models," in *EMNLP*, vol. 3, 2013, Conference Proceedings, p. 413.

[6] Z. Cao, F. Wei, L. Dong, S. Li, and M. Zhou, "Ranking with recursive neural networks and its application to multi-document summarization," in *AAAI*, 2015, Conference Proceedings, pp. 2153–2159.

[7] S. Zelikovitz and H. Hirsh, "Using lsi for text classification in the presence of background text," in *Proceedings of the tenth international conference on Information and knowledge management*. ACM, 2001, Conference Proceedings, pp. 113–118.

[8] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *European conference on machine learning*. Springer, 1998, Conference Proceedings, pp. 137–142.

[9] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[10] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[11] C. N. dos Santos and M. Gatti, "Deep convolutional neural networks for sentiment analysis of short texts," in *COLING*, 2014, Conference Proceedings, pp. 69–78.

[12] B. Hu, Z. Lu, H. Li, and Q. Chen, "Convolutional neural network architectures for matching natural language sentences," in *Advances in Neural Information Processing Systems*, 2014, Conference Proceedings, pp. 2042–2050.

[13] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.

[14] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in Neural Information Processing Systems*, 2015, Conference Proceedings, pp. 649–657.

[15] D. Tang, B. Qin, and T. Liu, "Document modeling with gated recurrent neural network for sentiment classification," *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1422–1432, 2015.

[16] K. S. Tai, R. Socher, and C. D. Manning, "Improved semantic representations from tree-structured long short-term memory networks," *arXiv preprint arXiv:1503.00075*, 2015.

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

[18] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," *arXiv preprint arXiv:1502.03044*, vol. 2, no. 3, p. 5, 2015.

[19] A. M. Rush, S. Chopra, and J. Weston, "A neural attention model for abstractive sentence summarization," *arXiv preprint arXiv:1509.00685*, 2015.

[20] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, Conference Proceedings.

[21] J. Duncan and G. W. Humphreys, "Visual search and stimulus similarity," *Psychological review*, vol. 96, no. 3, p. 433, 1989.

[22] J. L. Elman, "Finding structure in time," *Cognitive science*, vol. 14, no. 2, pp. 179–211, 1990.

[23] B. Pang and L. Lee, "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*. Association for Computational Linguistics, 2005, Conference Proceedings, pp. 115–124.

[24] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, vol. 1631. Citeseer, 2013, Conference Proceedings, p. 1642.

[25] B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," in *Proceedings of the 42nd annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2004, Conference Proceedings, p. 271.

[26] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, 2011, Conference Proceedings, pp. 142–150.

[27] A. Graves, "Generating sequences with recurrent neural networks," *arXiv preprint arXiv:1308.0850*, 2013.

[28] R. Socher, B. Huval, C. D. Manning, and A. Y. Ng, "Semantic compositionality through recursive matrix-vector spaces," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, 2012, Conference Proceedings, pp. 1201–1211.

[29] S. I. Wang and C. D. Manning, "Fast dropout training," in *ICML (2)*, 2013, Conference Proceedings, pp. 118–126.

[30] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," *arXiv preprint arXiv:1404.2188*, 2014.

[31] P. Liu, X. Qiu, and X. Huang, "Recurrent neural network for text classification with multi-task learning," *arXiv preprint arXiv:1605.05101*, 2016.