Future Generation Computer Systems 113 (2020) 571–578

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Attention-based sentiment analysis using convolutional and recurrent neural network

Mohd Usama [a], Belal Ahmad [a], Enmin Song [a,*], M. Shamim Hossain [b], Mubarak Alrashoud [b], Ghulam Muhammad [c]

[a] School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China
[b] Chair of Smart Cities Technology, and Department of Software Engineering, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia
[c] Department of Computer Engineering, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia

ARTICLE INFO

Article history:
Received 23 December 2019
Received in revised form 10 June 2020
Accepted 8 July 2020
Available online 10 July 2020

Keywords:
CNN
RNN
Attention mechanism
Sentiment analysis

ABSTRACT

Convolution and recurrent neural network have obtained remarkable performance in natural language processing(NLP). Moreover, from the attention mechanism perspective convolution neural network(CNN) is applied less than recurrent neural network(RNN). Because RNN can learn long-term dependencies and gives better results than CNN. But CNN has its own advantage, can extract high-level features by using its local fix size context at the input level. Thus, this paper proposed a new model based on RNN with CNN-based attention mechanism by using the merits of both architectures together in one model. In the proposed model, first, CNN learns the high-level features of sentence from input representation. Second, we used attention mechanism to get the attention of the model on the features which contribute much in the prediction task by calculating the attention score from features context generated from CNN filters. Finally, these features context from CNN with attention score are commonly used at the RNN to process them sequentially. To validate the model we experiment on three benchmark datasets. Experiment results and their analysis demonstrate the effectiveness of the model.

1. Introduction

The sentiment analysis from text is an essential area of research in NLP. Its facilitates the human language to interact with computers and to understand reception of people such as their emotions and thinking about any product or service based on their opinions or discussion. Nowadays, analysis of opinions in form of short text getting much attention of the researchers because enormous amount of generated data from internet web, smart gadgets, and social network in the form of short text which contains people emotions and opinions about services, product, and movies are available. Analysis of these short text is an interesting research problem cause its helps companies and seller to understand the thinking of buyers and users of the products and services, whether they like their products and services or not. In general whether their products and services are doing good in market or not. The task of sentiment analysis is to understand the opinion and emotion attached to the text and assign them predefined domain automatically where its belong.

In early days many researchers have done sentiment analysis task by using traditional classification approaches which are based on manual features engineering and required rule-based methods such as data mining technique [1], knowledge-based approach [2]. But these traditional approaches mainly used words as a statistical indicator representation such as TF–IDF (Term frequency–Inverse document frequency). These schemes do not consider words feature as a positional factor. Usually, it does not represent the feature which has large weights. Besides this, some researcher used machine learning algorithms [3] for feature extraction and sentiment classification. However, these algorithms faced encounter problems like data sparsification during small training dataset.

Later after increasing in training datasets, researchers started to used distributed representation methods with deep learning approaches [4] which do not require manual features engineering and can deal with the above problems. These approaches automatically extract the features from text and perform appropriate analysis for many different problems in NLP. Deep learning approaches have successfully implemented in many areas of NLP including disease prediction [5] and emotion detection [6]. It

* Corresponding author.
E-mail addresses: mohdusama@hust.edu.cn (M. Usama), ahmadbelal@hust.edu.cn (B. Ahmad), esong@hust.edu.cn (E. Song), mshossain@ksu.edu.sa (M.S. Hossain), malrashoud@ksu.edu.sa (M. Alrashoud), ghulam@ksu.edu.sa (G. Muhammad).

https://doi.org/10.1016/j.future.2020.07.022
0167-739X/

is not only improving the accuracy and efficiency of prediction but also reducing the prediction time. Furthermore, traditional deep learning approaches such as CNN [7] and RNN [8] used max pooling layer to find important significant features of words. This max pooling layer approach could most likely help in text prediction and classification task to get better results. However, there was no specified way to deal with significant features; max pooling layer is selected the significant features corresponding to maximum activation value.

Thus to deal with this issue in more significant way idea of attention mechanism in deep learning is first ever proposed by Bahdanau et al. [9] for machine translation. Attention mechanism is a way to get the concentration of the model on the significant features by using softmax at the interior of the model. Generally, softmax is used as an output system to generate the probability of the classes and categories. However, It is also possible to use softmax in a way to normalized a stack of the number, so they all sum up to one; which effectively giving a percentage probability to all piece of information. Then it can use that to bring way during input data. More specifically, we know that not all word features in the sentence representation contribute equally in the text. So, here we use CNN-based attention approach to gain the attention of the model on such word features which contribute to the meaning in the text.

Attention-based neural networks have been proven successful in many problems of NLP including disease diagnosis [10]. However, most of the work in the attention mechanism is based on RNN and its variants. Attention-based RNN used three states of inputs to evaluates results at current states, i.e., the current input is given to RNN, recurrent input, and attention score. After the success of attention mechanism, significant work has also done on CNN with attention mechanism to solve different problem in NLP. However, CNN has been proven beneficiary in both ways, with and without attention mechanism. For sentence modeling where CNN learns word representation without attention [11] and sentence modeling with attention [12].

Our motivation of using RNN with CNN-based attention mechanism is inspired by the individual success of CNN and RNN with attention mechanism in NLP.

This study proposed a new model based on RNN with CNN-based attention mechanism by using the merits of both architectures together in one model. First, we used CNN to learn high-level features context from input representation. Then over these features context, we used attention mechanism to get the attention score. Finally both features context from CNN and attention score are commonly used as an input to RNN. Then final feature map from RNN is used to perform sentiment analysis task.

To evaluate the proposed model, we experiment with four self variation of the model on three benchmark datasets. The detail of the model's variation is given in Section 4.2. Experiment results show the effectiveness of the model. Our model achieves state-of-the-art results on two out of the three benchmark.

The contributions of article are as follows:

- Put forward a new model based on RNN with CNN-based attention mechanism for sentiment analysis.
- Furthermore, demonstrate four self variations of the proposed model i.e. AttConv RNN-pre, AttPooling RNN-pre, AttConv RNN-rand, and AttPooling RNN-rand. Experiment results exhibit that the AttPooling RNN-pre and AttPooling RNN-rand performs better than AttConv RNN-pre and AttConv RNN-rand, respectively.
- Apply all model variations on three benchmark datasets (Movie Reviews, Stanford Sentiment Treebank, and Treebank2) to test the effectiveness of the proposed models and achieves better results on two out of three datasets.

The remaining article is categorized in the following way. Section 2 explains the existed relevant work. Section 3 explains the proposed model architecture. Section 4 explains the model variations, experiment setup, and benchmark datasets used to measure the performance of the models. Section 5 discusses the analysis of experiment results. Finally, Section 6 concludes the paper.

2. Related work

2.1. Sentiment analysis task

Many works have done in sentiment classification by using several methods including data mining technique [13], knowledge-based approach [14], and machine learning algorithms [15,16]. Later with the rapid growth of deep learning, CNN and RNN algorithms achieved notable success in NLP [17]. Yoon Kim [18] used CNN with pre-trained vectors for classification of sentences. Socher et al. [19] proposed recursive tensor model for semantic compositionality. Tai et al. [20] proposed a LSTM model for semantic relatedness prediction and sentiment classification. Liu et al. [21] proposed attention-based gated CNN for sentiment analysis. Besides this, they proposed a new activation function named them NLReLU.

Some of the sentiment classification works have done by using two network architecture together in one model. Kim et al. [22] proposed a framework for modeling language by using LSTM, CNN, and highway network together. Furthermore, Hassan et al. [23] proposed combine CNN-LSTM architecture for sentence classification. Chen et al. [24] proposed DeepNetQoE for emotion recognition and and sentiment analysis with a balance between resources and model performance while achieving satisfactory training results. Usama et al. [25] proposed multi-architecture based feature fusion approach for sentiment analysis. Ren et al. [26] put forward a model named "lexicon-enhanced attention network (LEAN)" by using bidirectional LSTM. The proposed model not only able to find the sentiment word but also focus on aspect information in the sentence.

2.2. Attention-based models

In CNN, significant work was done based on attention. Gehring et al. [27] proposed the attention-based CNN model for machine translation. They used a hierarchical convolutional layer for both encoder and decoder. Each encoder hidden units be queried at nth decoder layer by output hidden units of convolution, after than weighted sum of the total encoder hidden units will be added to the decoder hidden units, and these updated hidden units will be received by $n + 1$ convolutional layer. Hence, the above attention-based model needs the multi convolutional layer to pass the weighted context from encoder to decoder side, which shows that their attention played a role after convolution. There are some other CNN architecture [28,29] which used attention at the pooling layer and named it attentive pooling. The architecture used in both paper uses two input sentence in parallel and sets of the hidden unit for every sentence is computed by convolution. Thus, every sentence learns attentive weights for each hidden unit based on the current hidden unit with all hidden unit in other sentences. After that resulted representation of each sentence will obtain by weighted mean pooling or attentive pooling.

The foremost RNN-based attention model was proposed by Graves et al. [30]. They used differentiable attention schemes which enforce RNN to look over distinct parts of the input. This scheme explored by many researchers to solve the problem of text processing such as text generation [31]. Gazpio et al. [32] proposed a self-attention based supervised model to learn word

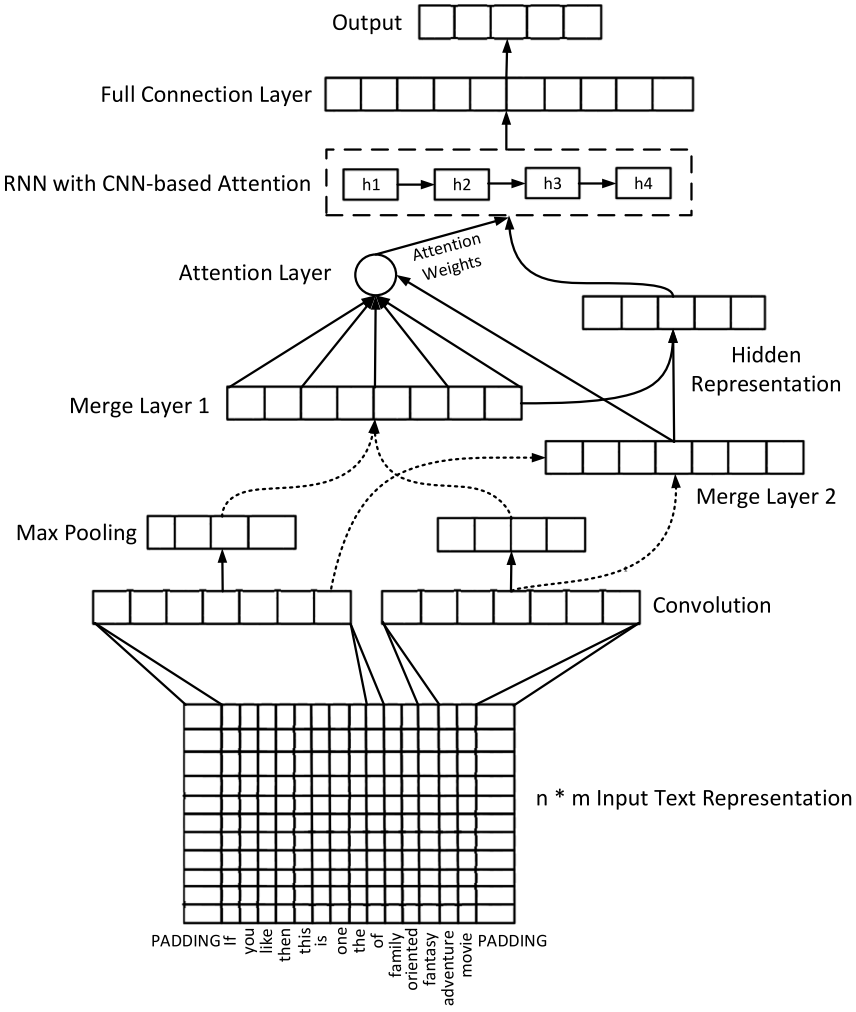

Fig. 1. Combine architecture of proposed framework. Dotted arrow line indicate that either features context will be passed from convolution to merge layer 2 or from max pooling to merge layer 1.

embedding. This method can calculate attention weight at each position in the window for every word. A sentence level attention is proposed by Liu et al. [33]. They used mean pooling as an attention context over LSTM states and re-weighted pooling vector of the sentence. Li et al. [34,35] proposed similar self-attention based factoid model for question encoding. Cheng et al. [36] proposed LSTMN model which is intra-sentence based attention scheme and further used by Parikh et al. [37]. Proposed scheme calculates the attention vector during the recurrent process for every hidden unit, which is targeted on lexical correlations between adjacent words. Pergola et al. [38] proposed a hierarchical model for sentiment classification and topic extraction. Their model extract aspect-sentiment clusters without using aspect level annotation. Liang et al. [39] proposed sequence-to-sequence selective attention-based model for summarization of social text. They used reinforcement learning and cross-entropy for the ROUGE score optimization directly, and used specific gate to filter invalid information.

In contrast to these, our proposed model uses the RNN with CNN-based attention mechanism. In the proposed model, RNN used feature's context and attention score generated from standard convolution or max pooling together as an input.

3. Proposed architecture design

This section describes the proposed architecture including Input layer, embedding layer, CNN layers, CNN-based attention, RNN layer with CNN-based attention mechanism, full connection layer, and output layer. The overall architecture of the proposed model is shown in Fig. 1.

3.1. Input layer

To start let assume the input layer receives text data as $X(x_1, x_2, \ldots, x_n)$, where $x_1, x_2, \ldots, x_n$ are the n number of word with dimension of each input word m. So, each word vector will be represented as R^m dimensional space. Therefore, R^{m*n} will be the dimension space of input text.

3.2. Word embedding layer

To learn word embedding let assume size of the vocabulary is d for text represent. Thus, the 2-dimensional word embedding matrix would be represented as A^{m*d}. Now, the input text $X(x_i)$, Where $i = 1, 2, 3, \ldots, n.$, $X \epsilon R^{m*n}$, is pass to embedding layer from the input layer to generate word embedding vector from pure corpus text. We use a distributed representation method to learn word embedding and implement it through the word2vec model for pre-trained model variation.

Hence, actual input text to be feed into the model would be the representation of input text $X(x_1, x_2, \ldots, x_n) \epsilon R^{m*n}$ as numerical word vectors. Where, $x_1, x_2, ..x_n$ are n number of word vectors in embedding vocabulary with each dimension space R^m.

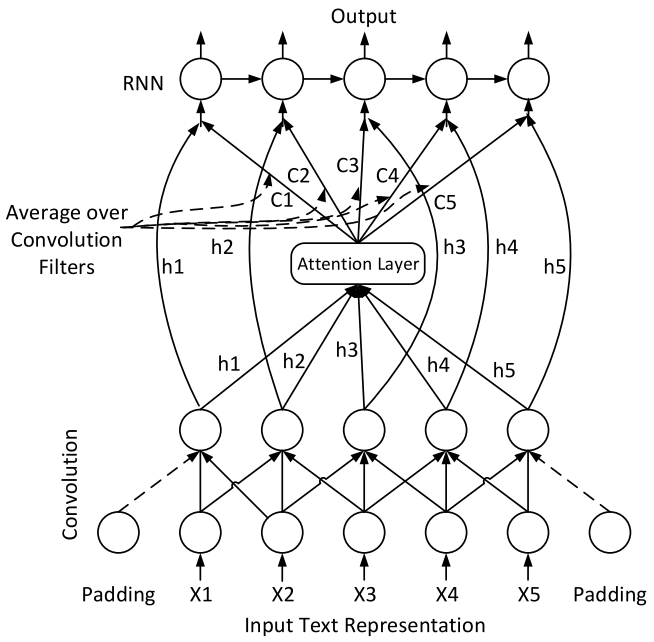

Fig. 2. Illustration of features context and attention score together used at RNN. X_i are the input representations and C_i are attention score calculated based on features context h_i generated from standard convolution.

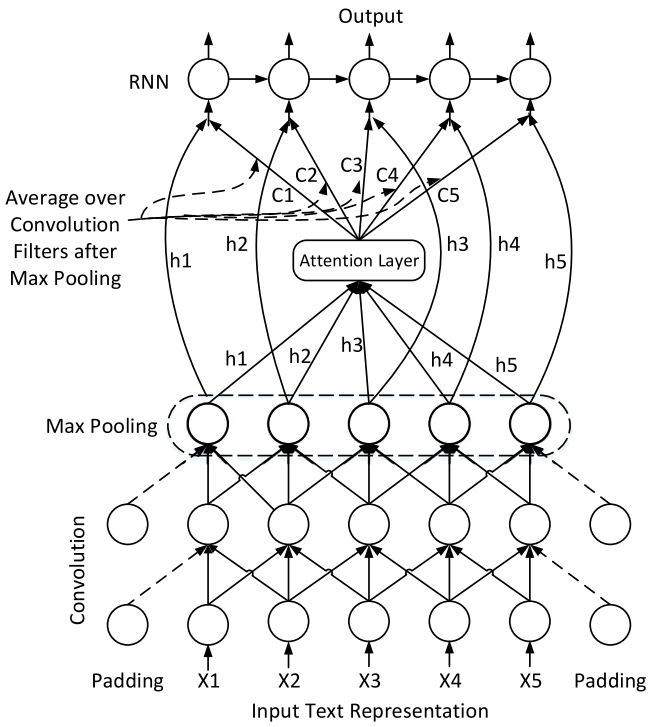

Fig. 3. Illustration of features context and attention score together used at RNN. X_i are the input representations and C_i are attention score calculated based on features context h_i generated from max pooling.

Where dimension of input word m will be selected during word embedding learning.

3.3. CNN layers

3.3.1. Convolution layer

After receiving text representation from embedding layer, convolution operation is performed over it in row representation form. Let s word vectors are selected at a time to perform convolution operation with weight matrix $W \epsilon R^{s*m}$ as follows:

$$h_i = f(X_{i+s-1} * W[i] + b_i) \tag{1}$$

Here, f is Relu function for non-linearity, $h_i \epsilon R^{n-s+1}$ is generated features context by having s word vectors in convolution operation every time repeatedly, and b_i is bias.

3.3.2. Max pooling layer

Now, Max-pooling operation is performed over features generated from convolution as follow:

$$p_i = \max h_i \tag{2}$$

Here, $p_i \epsilon R^{n-s+1/2}$ is feature map obtain after pooling operation.

We take the experience from the literature [18] and use two convolution layer in parallel with filter size 4 and 5. Then, we concatenate features of both convolution layers at merge layer 2 and after performing pooling operation at merge layer 1 as shown in Fig. 1. We also perform flattening after merge layer 1 and 2 to convert the convolved and pooled features to a single column that is passed to RNN.

3.4. CNN-based attention

After obtaining the features context $h_1, h_2, \ldots, h_n$ from CNN, we will calculate the attention score for each feature context based on all features context. Now, we know that attention requires all hidden units as input to compute attention vector. Thus attention score C_i for each feature context h_i is calculated as weighted sum over all features context, as follows:

$$C_i = \frac{exp(h_i)}{\sum_{i.} exp(h_{i.})} \tag{3}$$

Here for AttConv RNN, attention score is calculated by averaging features context generated from convolution layer as shown in Fig. 2. For AttPooling RNN, attention score is calculated by averaging features context generated from max pooling layer as shown in Fig. 3.

This attention score is used with the RNN and jointly learned during the training process. This attention score is expected to focus on such features context which play an essential role by contributing meaning to the text in sentiment prediction.

3.5. RNN layer with CNN-based attention

We know that RNN with gated mechanism i.e. LSTM is able to handling log-term dependency in sentence. Thus here we used LSTM; a powerful variant of RNN. Now, after computation of attention score, we will use features context h_i from CNN and attention score C_i together at RNN to generates the final feature map by processing them sequentially.

Hence, the final feature map will be obtain by jointly learning of features context from CNN and the attention score C_i(as shown in Fig. 1) as follows:

$$h(t)_i^1 = LSTM(h(t)_i, h(t-1)_i, C(t)_i) \tag{4}$$

where, $h(t)$ is the hidden representation at time-step t, $h(t-1)$ is the hidden representation at time-step $t-1$, C_i is the attention score, and $h^1(t)$ is the final feature map obtain from RNN. Here we use location based attention approach by averaging CNN-based features context.

3.6. Full connection layer

After that, we use the full connection layer. The full connection operation will be perform over feature map $h(t)_i^1$ obtain from RNN as follows:

$$h^2 = w^1.h^1 + b^1 \tag{5}$$

where h^1 is a feature map receive from RNN and h^2 is feature map obtain from full connection operation. w^1 and b^1 are the weight and bias of full connection layer.

3.7. Output layer

Finally, output layer will perform the classification of sentiment using the features from the full connection layer, as shown in Fig. 1, Here, Sigmoid and Softmax classifier are used for binary and multiclass dataset respectively. Cross-entropy is used to compute the discrepancy between predicted and actual sentiment of the text.

4. Experiment setup, model variations, and datasets

4.1. Training procedure and hyper-parameters setting

The hyper-parameters values selected for all dataset are as follows: activation function as Relu, convolution filter size 4 and 5, CNN layers hidden size 100, RNN output size 100, dropout 0.50, and mini-batch size 50. Adadelta update optimizer is used to update the model's parameters during training. The hyper-parameters values are selected through the random-search method. For word embedding we used pre-trained set of vectors obtain from word2vec[1] with 300 dimension.

4.2. Summary of model variations

Four model variations are tested in experiment, defined as follows:

- **AttConv RNN-pre:** A model based on RNN with convolution-based attention mechanism trained with pre-trained word-vectors from word2vec. Here attention score is calculated by averaging the features context generated from standard convolution layer as shown in Fig. 2.
- **AttPooling RNN-pre:** A model based on RNN with pooling-based attention mechanism trained with pre-trained word-vectors from word2vec. Here attention score is calculated by averaging the features context generated from max pooling layer as shown in Fig. 3.
- **AttConv RNN-rand:** A model based on the RNN with convolution-based attention mechanism trained with randomly initialized word-vectors. Here attention score is calculated by averaging the features context generated from standard convolution layer as shown in Fig. 2.
- **AttPooling RNN-rand:** A model based on RNN with pooling-based attention mechanism trained with randomly initialized word-vectors. Here attention score is calculated by averaging the features context generated from max pooling layer as shown in Fig. 3.

[1] https://code.google.com/p/word2vec/.

Table 1
Detail statistics after tokenization. c: Number of classes, A_l:Average sentence length, M_l:Maximum sentence length, V:Dataset size, N:Vocabulary size, V_{Train}:Training set, V_{Test}:Test set, V_{Valid}:Validation set, CV (10-fold Cross validation) is used when there is no standard split available for train, test, and val set.

Dataset	c	A_l	M_l	V	N	V_{Train}	V_{Test}	V_{Valid}
MR	2	20	51	10 662	18 983	8655	1064	CV
SST1	5	18	56	11 855	18 784	8544	2210	1101
SST2	2	19	56	9613	18 784	6920	1821	872

4.3. Datasets

Three Benchmark datasets are used in experiment. Detail statistics of datasets are listed in Table 1.

- **MR[2](Movie Reviews):** This is a binary classified movie reviews dataset including one review per sentence [40].
- **SST1[3](Stanford Sentiment Treebank):** This is a multiclass dataset, classified into five categories as negative, very negative, positive, very positive, and neutral with provided standard split set into train, test, and deviation.
- **SST2** Subset of SST1 including binary labeled and removed neutral reviews.

5. Experiment results and discussion

Implementation of model architecture is done by using Tensorflow and Keras library. We compare the running time for model variations on the system having 16 GB GPU. Here, we execute all model variations up to 30 epochs to compare running time with each other. Fig. 4 shows the comparison of the running time for all models variation. Section 4.1 describes the parameter settings and training method of the model's parameters and word vectors. Experimentation is done using three datasets to evaluate the model performance; experiments result are mentioned in Table 2. As expected model's variant (AttPooing RNN-rand and AttConv RNN-rand) with randomly initialized word-vectors does not perform better than pre-trained model variations (AttPooing RNN-pre and AttConv RNN-pre). While expected results were achieved with pre-trained vectors. Even model's variant with pre-trained (AttConv RNN-pre) received better results than some NLP models which used complex structure [19]. The experiment results suggest that model with pre-trained vectors perform better compared to model with random initialize vectors. The accuracy gain of 2%–4% approximate can be achieved with pre-trained vectors compared to random initialize vectors.

Experiment results validate that the model performs better than state of the art methods on two out of three datasets, and achieve competitive results on one dataset. Especially, as we expected, our model AttPooling RNN-pre does perform better than AttConv RNN-pre over all the datasets. AttPooling RNN-pre achieves remarkable results and again in increase accuracy by 0.80% on MR, 0.73% on SST1, and 1.77% on SST2 dataset than AttConv RNN-pre. These results state that our architecture using attention approach over selected features by max pooling is more beneficiary than using attention over all features from convolution over all the three datasets.

[2] https://www.cs.cornell.edu/people/pabo/movie-review-data/.
[3] https://nlp.stanford.edu/sentiment/.

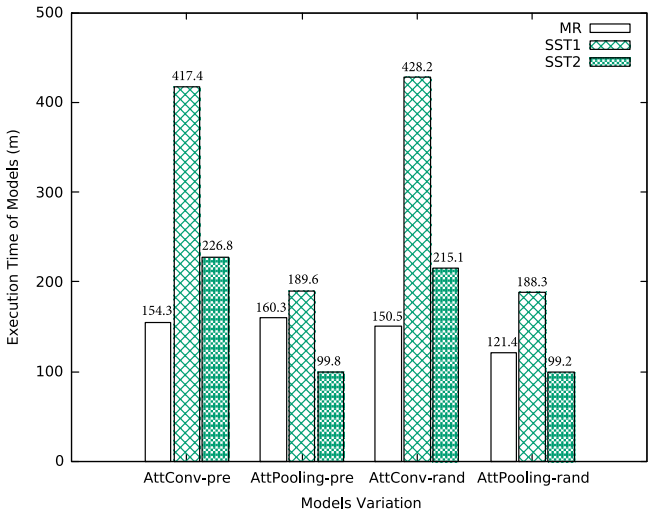

Fig. 4. Running time of all model variations. From these statistics, we can tell that there is not much difference in running time with pre-trained and randomly initialized models on all the datasets. That means AttPooling RNN-pre and AttConv RNN-pre consume approximate same time to complete execution as AttPooling RNN-rand and AttConv RNN-rand. While AttPooling RNN takes half of the times than AttConv RNN on SST1 and SST2 datasets for both randomly initialized and pre-trained. Thus running speed of AttPooing RNN is 2 times fast than AttConv RNN for both randomly initialized and pre-trained on SST1 and SST2 datasets. But for MR dataset running speed of all models are almost the same.

Table 2
The proposed model architecture results with baseline works. Bold represent the best results.

Models	Year	MR	SST1	SST2
Recursive Neural Tensor Network [19]	2013	–	45.7	85.4
Word vector averages [19]	2013	–	32.7	80.1
DCNN [11]	2014	–	48.5	86.8
Non-static [18]	2014	81.5	48.0	87.2
Multi-channel [18]	2014	81.1	47.4	88.1
Tree bi-LSTM [41]	2015	0.79	–	–
Tree-LSTM [42]	2015	–	48.0	–
LSTM [20]	2015	–	46.4	84.9
CNN-GRU-word2vec [43]	2016	82.28	50.68	**89.95**
Non-static GloVe+word2vec CNN [44]	2016	81.0	45.9	85.6
ConvLstm [45]	2017	–	47.5	88.3
CRDLM [23]	2018	–	48.8	89.2
AGCNN-SELU-3-channel [21]	2018	81.3	49.4	87.6
AGCNN-NLReLU-3-channel [21]	2018	81.9	49.4	87.4
ATTPooling RNN-pre	2019	**83.64**	**51.14**	89.62
ATTConv RNN-pre	2019	82.84	50.41	87.85
ATTPooling RNN-rand	2019	79.18	48.32	86.09
ATTConv RNN-rand	2019	77.66	46.36	84.90

5.1. Analysis of results on MR dataset

The challenge of this dataset is to predict sentiment from binary labeled sentence i.e. positive or negative which contain one sentence per review. Ref. [18] perform experimentation with several variations of the standard CNN model by doing fine tuning in word embedding and using a multichannel approach. Compared to his results our model accuracy is better and achieves 2% gain approximate. There is another work [44] which used bow SVM and same non-static CNN model as [18] but trained with combined Glove and Word2vec, and reported lesser accuracy than above. But compared to [44] we achieve 2.5% gain in accuracy approximate. Ref. [43] proposed a joint architecture by using CNN and RNN; reported much better results than previous works. This shows that combining CNN and RNN gives better results than single CNN or RNN architecture in the prediction task. The latest works [21] used attention with CNN and further achieve remarkable results which is better than existing works except for the joint architecture model. This latest work substantiates the benefits of attention in the prediction task. Thus in the proposed model we utilized CNN and RNN together with attention mechanism and achieve accuracy gain 1.74% than the latest CNN attention model [21] and 1.36% than joint architecture [43]. In overall, proposed model architecture is most modern of its type and achieves the better results among all existing works on MR dataset.

5.2. Analysis of results on SST1 dataset

SST1 dataset is fine-grain extension of MR dataset classified into five categories as negative, positive, very negative, very positive, and neutral. Early existing works on SST1 dataset based on different architectures such as RNTN [19], Deep CNN [11], Non-static and Multichannel CNN [18], LSTM [20], Tree-LSTM [42], and GloVe+word2vec CNN [44] reported the accuracy between 46% to 48% approximately except the Word vector averages [19]. Word vector average reported much worst results with accuracy 32.7%. Compared to early existing works our model achieves more than 2% gain in accuracy approximate. Later works such as CNN-GRU [43] and CRDLM [23] used joint architecture approach; reported 50.68% and 48.8% accuracy results respectively which is better compared to early existing works. But other joint architecture ConvLstm [45] could not achieve better results than all early existing works but still produced strong competitive results. Moreover, the latest work [21] based on attention mechanism with CNN reported poor results with accuracy 49.4%, than the joint architecture model but still get better results than early existing works. However, our model not only performs better than early existing works and joint architecture models but also than latest CNN based attention model [21]. The proposed model performs better than the latest existing models and achieves 0.46% and 1.74% accuracy gain than joint architecture [43] and gated-attention model [21] respectively. Finally we can say that our model architecture achieves state of the art result on SST1 dataset also.

5.3. Analysis of results on SST2 dataset

SST2 is a subset of SST1 with binary labeled and without neutral reviews. On SST2 dataset recursive and recurrent model such as RNTN [19] and LSTM [20] achieve accuracy between 84%–86% approximate. While all variation of standard CNN model (DCNN, Non-static CNN, and Multichannel CNN) achieve better results than existing recursive and recurrent model with accuracy more than 86% except GloVe+word2vec CNN [44] which got 85.6% accuracy. Compared to existing recursive, recurrent, and CNN models our model achieves better results with accuracy more than 89%. The proposed model also got gain in accuracy 2.02% when compared to latest gated-attention model [21]. But On this dataset joint architecture models (CNN-GRU [43] and CRDLM [23]) performed better compared to all existing models. However, compared to our model only one joint model (CNN-GRU) got little better results with 0.33% gain in accuracy. That means our model could not get state of the art results on SST2 but still giving better results than many existing works and a strong competitor of joint architecture models.

5.4. Comparative study with attention-based models

Moreover, we done comparative study with attention-based models to further measure the effectiveness of the our model architecture.

Significant works have done on attention-based CNN, and among them almost work in CNN is based on attentive pooling [29]. While significant amount of work has done on RNN and its variant with attention mechanism [30]. But the proposed model is based on RNN with CNN-based attention mechanism. In this study, the process of computing attentive context vectors is the similar to self-attention but it differs in following ways and have benefits over some attention based model define as follows: (i) Generally in self-attention approach, calculated attention score is used again with self architecture; but in the proposed model, attention score is calculated based on features context generated from CNN and then used again at RNN together with CNN's features context to learn the final representation of sentence. (ii) As we can see from Fig. 1, the proposed model is composed of both CNN and RNN, thus it gets benefits of both. CNN learns the robust local feature by using sliding convolution, and RNN learn long-term dependency by processing these feature sequentially with attention score generated from CNN itself.

6. Conclusion

This article proposed a new model architecture based on RNN with CNN-based attention for sentiment analysis task. The evaluation process of the model performed over three datasets. The model not only extract contextual and temporal features by CNN and RNN respectively but also focus on important feature maps by using attention scheme over features context generate from CNN. Analysis of results with baseline methods and comparative study demonstrates the effectiveness of the model. Moreover, performed experiments indicate that the model received better results than baseline methods on two out of three datasets with accuracy 83.64%, 51.14%, and 89.62% on MR, SST1, and SST2 dataset, respectively. The future study can be done to test same approach in other tasks of NLP.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

The authors are grateful to the Deanship of Scientific Research at King Saud University, Riyadh, Saudi Arabia for funding this work through the Vice Deanship of Scientific Research Chairs: Chair of Smart Cities Technology.

References

[1] S. Kim, E. Hovy, Automatic detection of opinion bearing words and sentences, in: The IJCNLP, 2005, pp. 61–66.
[2] S. Kim, E. Hovy, Automatic identification of pro and con reasons in online reviews, in: The COLING/ACL, 2006, pp. 483–490.
[3] M. Chen, Y. Cao, R. Wang, Y. Li, D. Wu, Z. Liu, DeepFocus: Deep encoding brainwaves and emotions with multi-scenario behavior analytics for human attention enhancement, IEEE Network. 33 (6) (2019) 70–77.
[4] M. Chen, P. Zhou, D. Wu, L. Hu, M. Hassan, A. Alamri, AI-skin: Skin disease recognition based on self-learning and wide data collection through a closed loop framework, Inf. Fusion 54 (2020) 1–9.
[5] Y. Zhang, X. Ma, J. Zhang, M.S. Hossain, G. Muhammad, S.U. Amin, Edge intelligence in the cognitive internet of things: improving sensitivity and interactivity, IEEE Network 33 (3) (2019) 58–64.
[6] M. Chen, Y. Hao, Label-less learning for emotion cognition, IEEE Transactions on Neural Networks and Learning Systems. 31 (7) (2020) 2430–2440.
[7] Yann LeCun, L'eon Bottou, Yoshua Bengio, Patrick Haffner, Gradient-based learning applied to document recognition, in: Proceedings of the IEEE, 1998, pp. 2278–2324.
[8] Jeffrey L. Elman, Finding structure in time, Cogn. Sci. 14 (2) (1990) 179–211.
[9] Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio, Neural machine translation by jointly learning to align and translate, in: Proceedings of ICLR, 2015.
[10] M. Usama, B. Ahmad, W. Xiao, et al., Self-attention based recurrent convolutional neural network for disease prediction using healthcare data, Comput. Methods Programs Biomed. https://doi.org/10.1016/j.cmpb.2019.105191.
[11] Nal Kalchbrenner, Edward Grefenstette, Phil Blunsom, A convolutional neural network for modelling sentences, in: Proceedings of ACL, 2014, pp. 655–665.
[12] Meng Joo Er, Zhang Yong, Wang Ning, Mahardhika Pratama, Attention pooling-based convolutional neural network for sentence modeling, Inform. Sci. 373 (2016) 388–403, http://dx.doi.org/10.1016/j.ins.2016.08.084.
[13] M.S. Hossain, et al., Audio-visual emotion-aware cloud gaming framework, IEEE Trans. Circuits Syst Video Technol. 25 (12) (2015) 2105–2118.
[14] Duc-Hong Pham, Anh-Cuong Le, Learning multiple layers of knowledge representation for aspect based sentiment analysis, Data Knowl. Eng. 114 (2018) 26–39, http://dx.doi.org/10.1016/j.datak.2017.06.001.
[15] Q. Fang, et al., Relational user attribute inference in social media, IEEE Trans. Multimedia 17 (7) (2015) 1031–1044.
[16] M. Chen, Y. Jiang, Y. Cao, A. Zomaya, Creativebioman: Brain and body wearable computing based creative gaming system, IEEE Syst. Man Cybern. Mag. 6 (1) (2020) 14–22.
[17] T. Young, D. Hazarika, S. Poria, E. Cambria, Recent trends in deep learning based natural language processing [Review article], IEEE Comput. Intell. Mag. 13 (3) (2018) 55–75, http://dx.doi.org/10.1109/MCI.2018.2840738.
[18] Yoon Kim, Convolutional neural networks for sentence classification, in: Proceedings of EMNLP, 2014, pp. 1746–1751.
[19] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. Manning, A. Ng, C. Potts, Recursive deep models for semantic compositionality over a sentiment treebank, in: Proceedings of EMNLP, 2013, pp. 16–42.
[20] K. Tai, R. Socher, C. Manning, Improved semantic representations from tree-structured long short-term memory networks, 2015, arXiv preprint arXiv:1503.00075.
[21] Yang Liu, Lixin Ji, Ruiyang Huang, Tuosiyu Ming, Chao Gao, Jianpeng Zhang, An attention-gated convolutional neural network for sentence classification, 2018, arXiv preprint arXiv:1808.07325.
[22] Y. Kim, Y. Jernite, D. Sontag, A. Rush, Character-aware neural language models, 2015, arXiv preprint arXiv:1508.06615.
[23] A. Hassan, A. Mahmood, Convolutional recurrent deep learning model for sentence classification, IEEE Access 6 (2018) 13949–13957, http://dx.doi.org/10.1109/ACCESS.2018.2814818.
[24] R. Wang, M. Chen, N. Guizani, Y. Li, H. Gharavi, K. Hwang, DeepNetQoE: Self-adaptive QoE optimization framework of deep networks, IEEE Network. (2020) arXiv preprint arXiv:2007.10878.
[25] M. Usama, W. Xiao, B. Ahmad, J. Wan, M.M. Hassan, A. Alelaiwi, Deep learning based weighted feature fusion approach for sentiment analysis, IEEE Access 7 (2019) 140252–140260.
[26] Z. Ren, G. Zeng, L. Chen, Q. Zhang, C. Zhang, D. Pan, A lexicon-enhanced attention network for aspect-level sentiment analysis, IEEE Access 8 (2020) 93464–93471, http://dx.doi.org/10.1109/ACCESS.2020.2995211.
[27] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, Yann N. Dauphin, Convolutional sequence to sequence learning, in: Proceedings of ICML, 2017, pp. 1243–1252.
[28] Wenpeng Yin, Hinrich Schutze, Bing Xiang, Bowen Zhou, ABCNN: Attention-based convolutional neural network for modeling sentence pairs, in: TACL, 2016, pp. 259–272.
[29] Cicero Nogueira dos Santos, Ming Tan, Bing Xiang, Bowen Zhou, Attentive pooling networks, 2016, CoRR, abs/1602.03609.
[30] Alex Graves, Greg Wayne, Ivo Danihelka, Neural turing machines, 2014, CoRR, abs/1410.5401.
[31] H. Zheng, W. Wang, W. Chen, A.K. Sangaiah, Automatic generation of news comments based on gated attention neural networks, IEEE Access 6 (2018) 702–710, http://dx.doi.org/10.1109/ACCESS.2017.2774839.
[32] I. Lopez-Gazpio, M. Maritxalar, M. Lapata, E. Agirre, Word n-gram attention models for sentence similarity and inference? Expert Syst. Appl. 132 (2019) 1–11, http://dx.doi.org/10.1016/j.eswa.2019.04.054.
[33] Yang Liu, Chengjie Sun, Lei Lin, Xiaolong Wang, Learning natural language inference using bidirectional LSTM model and inner-attention, 2016, CoRR, abs/1605.09090.

[34] Peng Li, Wei Li, Zhengyan He, Xuguang Wang, Ying Cao, Jie Zhou, Wei Xu, Dataset and neural recurrent sequence labeling model for open-domain factoid question answering, 2016, arXiv preprint arXiv:1607.06275.
[35] M. Chen, J. Zhou, G. Tao, J. Yang, L. Hu, Wearable affective robot, IEEE Access 6 (2018) 64766–64776.
[36] Jianpeng Cheng, Li Dong, Mirella Lapata, Long short-term memory-networks for machine reading, in: Conference on EMNLP, 2016, pp. 551–561.
[37] Ankur P. Parikh, Oscar Tackstrom, Dipanjan Das, Jakob Uszkoreit, A decomposable attention model for natural language inference, in: Proceedings of EMNLP, 2016.
[38] Gabriele Pergola, Lin Gui, Yulan He, TDAM, A topic-dependent attention model for sentiment analysis, Inf. Process. Manage. (ISSN: 0306-4573) 56 (6) (2019) 102084.
[39] Zeyu Liang, Junping Du, Chaoyang Li, Abstractive social media text summarization using selective reinforced seq2seq attention model, Neurocomputing (ISSN: 0925-2312) (2020).
[40] B. Pang, L. Lee, Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales, in: Proceedings of ACL, 2005.
[41] Jiwei Li, Dan Jurafsky, Eudard Hovy, When are tree structures necessary for deep learning of representations, arXiv preprint arXiv:1503.00185.
[42] X. Zhu, P. Sobhani, H. Guo, Long short-term memory over tree structures, 2015, arXiv preprint arXiv:1503.04881.
[43] Xingyou Wang, Weijie Jiang, Zhiyong Luo, Combination of convolutional and recurrent neural network for sentiment analysis of short texts, in: Proceedings of COLING 2016, pp. 2428–2437.
[44] Y. Zhang, B. Wallace, A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification, 2016, arXiv preprint arXiv:1510.03820.
[45] A. Hassan, A. Mahmood, Deep learning approach for sentiment analysis of short texts, in: Proceedings of the ICCAR, 2017.

Mohd Usama received B.Sc(Hons) degree in Statistics and Master degree in Computer Science from Aligarh Muslim University, India in 2012 and 2016 respectively. He is currently pursuing Ph.D. degree from the School of Computer Science and Technology, Huazhong University of Science and Technology(HUST), Wuhan, China since 2016. His research interests include deep learning, natural language processing, and healthcare Informatics.

Belal Ahmad received B.Sc(Hons) degree from Department of Statistics and Operation Research, Aligarh Muslim University(AMU), Aligarh, India in 2009. He received his MCA degree from Department of Computer Science, AMU, Aligarh, India in 2013. He is currently pursuing Ph.D. degree from the School of Computer Science and Technology, HUST, Wuhan, China. His research interests include network security and machine learning.

Enmin Song is currently a Professor at School of Computer Science and Technology, Huazhong University of Science and Technology, China. He received his Ph.D. degree in Electrical Engineering and Computer from the Teesside University, UK. After completing his Ph.D., he was postdoctoral researcher at University of California San Francisco (UCSF). He is a senior member of IEEE. His current research interests involve medical image processing and medical image information analysis.

M. Shamim Hossain is a Professor at the Department of Software Engineering, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia. He is also an adjunct professor at the School of Electrical Engineering and Computer Science, University of Ottawa, Canada. He received his Ph.D. in Electrical and Computer Engineering from the University of Ottawa, Canada. His research interests include cloud networking, smart environment (smart city, smart health), AI, deep learning, edge computing, Internet of Things (IoT), multimedia for health care, and multimedia big data. He has authored and coauthored more than 260 publications including refereed journals, conference papers, books, and book chapters. Recently, he co-edited a book on "Connected Health in Smart Cities, published by Springer. He has served as cochair, general chair, workshop chair, publication chair, and TPC for over 12 IEEE and ACM conferences and workshops. Currently, he is the cochair of the 3rd IEEE ICME workshop on Multimedia Services and Tools for smart-health (MUST-SH 2020). He is a recipient of a number of awards, including the Best Conference Paper Award and the 2016 ACM Transactions on Multimedia Computing, Communications and Applications (TOMM) Nicolas D. Georganas Best Paper Award, and the 2019 King Saud University Scientific Excellence Award (Research Quality). He is on the editorial board of the IEEE Transactions on Multimedia, IEEE Multimedia, IEEE Network, IEEE Wireless Communications, IEEE Access, Journal of Network and Computer Applications (Elsevier), and International Journal of Multimedia Tools and Applications (Springer). He also presently serves as a lead guest editor of ACM Transactions on Internet Technology, IEEE network, Multimedia Systems, and IEEE Access. Previously, he served as a guest editor of IEEE Communications Magazine, IEEE Transactions on Information Technology in Biomedicine (currently JBHI), IEEE Transactions on Cloud Computing, International Journal of Multimedia Tools and Applications (Springer), Cluster Computing (Springer), Future Generation Computer Systems (Elsevier), Computers and Electrical Engineering (Elsevier), Sensors (MDPI), and International Journal of Distributed Sensor Networks. He is a senior member of both the IEEE, and ACM.

Mubarak Alrashoud received the Ph.D. degree in computer science from Ryerson University, Toronto, ON, Canada in 2015. He is currently an Associate Professor and the Head of the Department of Software Engineering, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia. He is member of IEEE.

Ghulam Muhammad is a full professor in the Department of Computer Engineering, College of Computer and Information Sciences at King Saud University (KSU), Riyadh, Saudi Arabia. Prof. Ghulam received his Ph.D. degree in Electrical and Computer Engineering from Toyohashi University and Technology, Japan in 2006, M.S. degree from the same university in 2003. He received his B.S. degree in Computer Science and Engineering from Bangladesh University of Engineering and Technology in 1997. He was a recipient of the Japan Society for Promotion and Science (JSPS) fellowship from the Ministry of Education, Culture, Sports, Science and Technology, Japan. His research interests include image and speech processing, cloud and multimedia for healthcare, AI, and machine learning. Prof. Ghulam has authored and co-authored more than 250 publications including IEEE/ACM/Springer/Elsevier journals, and flagship conference papers. He owns two U.S. patents. He received the best faculty award of the Computer Engineering department at KSU during 2014–2015. He supervised more than 15 Ph.D. and Master Theses. Prof. Ghulam is involved in many research projects as a principal investigator (approximate amount of half a million US dollars) and a co-principal investigator (approximate amount of 1.5 million US dollars).