Turk J Elec Eng & Comp Sci
() : –
© TÜBİTAK
doi:10.3906/elk-

Sentiment classification using Attention based gated-CNN with deep recurrent neural model

S M Mushfiqur RAHMAN[1*], Ashmita RIYA[1,2], S M Rafizul HAQUE[1,3†]
[1]Computer Science & Engineering Discipline, Khulna University, Khulna, Bangladesh, ORCID iD: https://orcid.org/0000-0003-3417-9726
[2]Computer Science & Engineering Discipline, Khulna University, Khulna, Bangladesh, ORCID iD: https://orcid.org/0000-0001-5771-6470
[3]Computer Science & Engineering Discipline, Khulna University, Khulna, Bangladesh, ORCID iD: https://orcid.org/0000-0001-7757-261

Received: .201 • **Accepted/Published Online:** .201 • **Final Version:** ..201

Abstract: Sentiment analysis received a lot of attention recently due to its potential use in business intelligence. Understanding variable length sentences to extract the sentimental context is the main challenge of this concept. Our proposed models are moderations of a deep neural model named comprehensive attention recurrent model [5]. A new layer of attention mechanism and replacement of LSTM with gated-CNN have been introduced to make learning of CA model [5] faster and efficient. IMDB movie review sentiment-labelled dataset has been used in our experiments. Our paper solely focuses on the comparison of performances among proposed and inspired models. Experimental results imply that accuracy and precision of our proposed models are better compared to the state-of-the-art CA model.

Key words: Sentiment Classification, Recurrent Neural Network (RNN), Long Short-term Memory (LSTM), Gated Recurrent Unit (GRU), Convolutional Neural Network (CNN), Comprehensive Attention, Attention Mechanism, Gated-CNN

1. Introduction

Sentiment Classification has emerged as a significant research topic of natural language processing (NLP). It is the task of classifying texts according to the sentiments in the form of positive, negative or neutral [1]. Sentiment classification measures accuracy on how well it matches with the human judgment, e.g., "Avengers: Infinity war is the best marvel movie so far" is a positive judgment according to human perspective and it should be predicted as a positive judgment as well. Machine learning techniques such as maximum entropy [2], support vector machine (SVM) [3] were used in sentiment classification to achieve clusters of similar sentiment.

Deep neural network models have been proven to be powerful text classifiers. Our proposed models are based on a framework named Comprehensive Attention Recurrent Neural Network (CA-RNN) [5]. We placed new layer named attention mechanism with CA models. Also we tried introducing gated-CNN. Our objective is to find out the best moderated model from all the proposed models. word2vec with Skip-gram method is used for

*Correspondence: elektrik@tubitak.gov.tr
†Correspondence: elektrik@tubitak.gov.tr

This work is licensed under a Creative Commons Attribution 4.0 International License.

word embedding [19] to get context of all the words. It takes a data corpus as input and creates a vector space of numbers of several dimensions. Finally, a sigmoid function is used for sentiment labeling. In this paper, we apply attention mechanism with CA-LSTM and CA-GRU to compare performance of the proposed models against the existing models. Figure 1 demonstrates the whole procedure of our model.

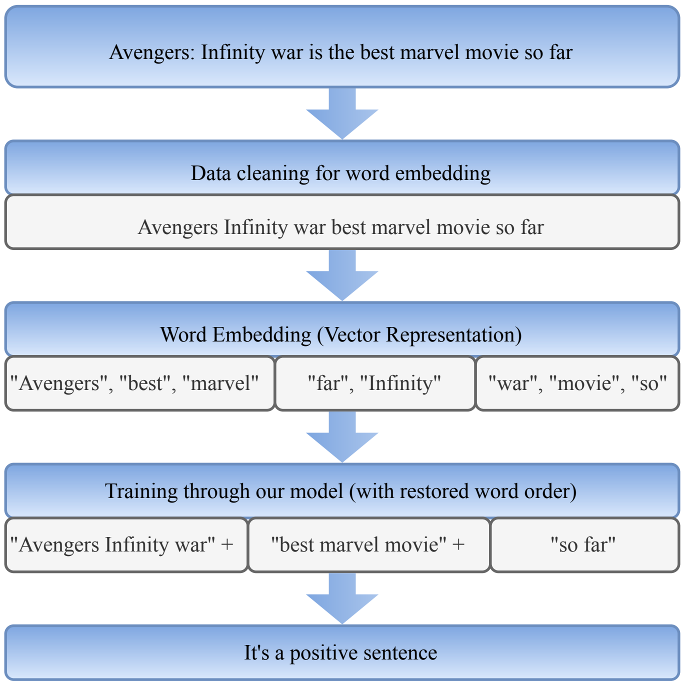

Figure 1: Procedure of sentiment classification in our model.

This paper is organized as follows: Section 2 gives a review of the related works regarding our proposed models. In Section 3, we have discussed about several neural network models that are used in our implementations. Section 4 demonstrates all the proposed models that have been used for classifying sentiment. Experimental results, dataset, precision-recall curve etc. are described in section 5. Conclusion and future works are stated in Section 6.

2. Background

Before implementing attention mechanism, the initial components of comprehensive attention model such as Bidirectional recurrent neural network (BRNN), long short-term memory (LSTM) and gated recurrent unit (GRU) are described in brief here.

2.1. Bidirectional Recurrent Neural Network (BRNN)

The neurons of a regular RNN, divided and directed into two directions, one in positive time direction (forward states) and another in negative time direction (backward states), results in BRNN [7]. The bidirectional recurrent neural network (BRNN) is considered to access both the preceding and succeeding contexts of input by combining a forward hidden layer $\vec{h}$ and a backward hidden layer $\overleftarrow{h}$ as illustrated in Figure 2. The working

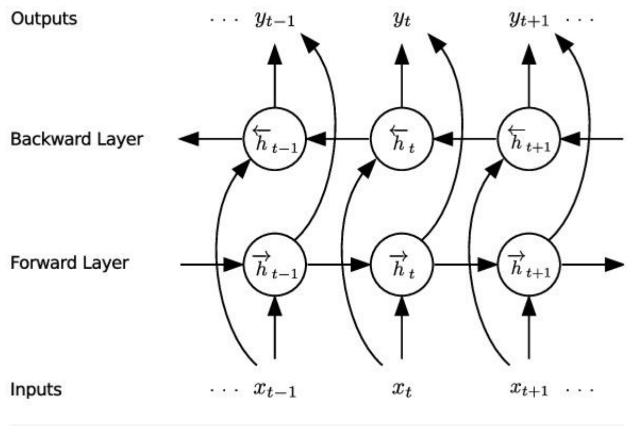

Figure 2: A BRNN model illustration

principle of BRNN is similar to RNN. But while applying back-propagation, additional processes are required for updating input and output layers as it cannot be done at once [**?**]:

$$\vec{h}_t = g(W_{x\vec{h}}x_t + W_{\vec{h}\vec{h}}\vec{h}_{t-1} + b_{\vec{h}}) \tag{1}$$

$$\overleftarrow{h}_t = g(W_{x\overleftarrow{h}}x_t + W_{\overleftarrow{h}\overleftarrow{h}}\overleftarrow{h}_{t-1} + b_{\overleftarrow{h}}) \tag{2}$$

$$y_t = W_{\vec{h}y}\vec{h}_t + W_{\overleftarrow{h}y}\overleftarrow{h}_t + b_y \tag{3}$$

In case of BRNN, gradient (rate of cost changes with respect to weights and bias) is a major factor. While performing back-propagation for multiple hidden layers 'vanishing gradient' may occur due to multiplying gradient with previous gradients. To resolve this problem, LSTM and GRU models are introduced.

2.2. Long Short Term Memory (LSTM)

LSTM is a recurrent network that takes word in a sequence, at time t, it takes the t-th word and also output from time t-1 as input (Figure 3). The center piece of LSTM is memory cells c_t designed to counteract the risk of vanishing/exploding gradients, thus enabling learning of dependencies over larger time lags. The forget gate f_t is for resetting the memory cells. The input gate i_t and output gate o_t control the input and output of the memory cells [8]. The looping arrows indicate recursive nature of the cell to store the context. The hidden state h_t is computed with given input x_t as follows:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \tag{4}$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \tag{5}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \tag{6}$$

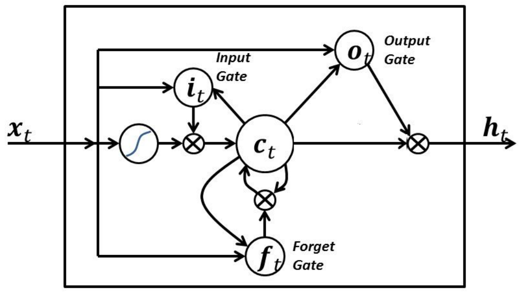

Figure 3: A LSTM unit illustration

$$\widetilde{c_t} = \tan h(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{7}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \widetilde{c_t} \tag{8}$$

$$h_t = o_t \odot \tan h(c_t) \tag{9}$$

where i is an input gate, modulating how much new memory content is added to the memory, f is the forget gate modulating how much existing memory is forgotten and o is an output gate modulating the amount of memory content. The memory cell c_t consists of two components, namely partially forgotten previous memory c_{t-1} and modulated new memory $\widetilde{c_t} \cdot \odot$ denotes element-wise multiplication and σ is an element wise squash function to make the gating values in [0, 1], but in case of sentiment classification it is [-1, 1]. The squash function includes sigmoid, hyperbolic tangent and ReLU.

2.3. Gated Recurrent Unit (GRU)

The gated recurrent unit (GRU) [9] is a modulated version of LSTM without a separate memory cell. It combines the forget gate and input gate into a single update gate. The GRU does not have an output gate; hence, it exposes its full hidden contents. A reset gate controls the flow of previous activations when computing new activations (Figure 4). The hidden state h_t, given input x_i is computed as follows:

$$z_t = \sigma(W_{xz}x_t + W_{hz}h_{t-1} + b_z) \tag{10}$$

$$r_t = \sigma(W_{xr}x_t + W_{hr}h_{t-1} + b_r) \tag{11}$$

$$\widetilde{h_t} = \tan h(W_{xh}x_t + W_{hh}(r_t \odot h_{t-1}) + b_h) \tag{12}$$

$$h_t = (1 - z_t)h_{t-1} + z_t\widetilde{h_t} \tag{13}$$

where z is the update gate and r is the reset gate. The gate values and hidden layer outputs also lie within the range [0, 1]. The working procedure of GRU in the hidden layer is almost same as LSTM.

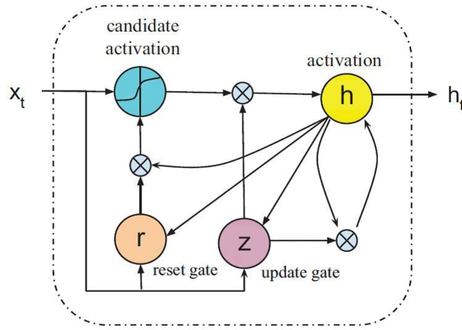

Figure 4: A GRU model illustration

2.4. Convolutional Neural Network (CNN)

Convolutional neural network is a deep feed-forward neural network. The architecture of a CNN includes input layer which takes a sequence of input c with n entities. Each entity is represented by d-dimensional dense vector. Thus, the input c is represented as a feature map of dimensionality $d * n$. Convolutional layer is used for representation learning from sliding w-grams [10]. Finally, max pooling is used to reduce the size of input. For an input sequence with n entries: c_1, c_2,.......,c_n, let vector $x_i \in R^{wd}$ be the concatenated embedding of w entries. Then the generation of the representation $L_{c_i} \in R^d$ for w-gram using the convolutional weights $W_c \in R^{d*wd}$ and bias $b \in R^d$ is:

$$L_{c_i} = g(W_c \cdot x_i + b) \tag{14}$$

CNN model holds the local context information from equation (14).

2.5. Comprehensive Attention Recurrent Neural Networks

The comprehensive attention recurrent models [5] uses bidirectional recurrent models (RNN/LSTM/GRU) that can access both historical and future contexts. A convolutional layer is to capture the local information in a sequence. In the comprehensive attention model, historical, local and future contexts are combined together. Suppose the length of a sentence is n. We set the border mode of convolution as “same” so that the output length of convolutional layer is the same as the input. As the word window slides the feature maps of the convolutional layer, it can be represented as follows:

$$H_c = [H_{c_1}, H_{c_2}, ..., H_{c_n}] \tag{15}$$

$$L_c = [L_{c_1}, L_{c_2}, ..., L_{c_n}] \tag{16}$$

$$F_c = [F_{c_1}, F_{c_2}, ..., F_{c_n}] \tag{17}$$

Here, H_c, L_c and F_c represent historical, local and future context representations respectively. The comprehensive attention of a sentence can be calculated as follows:

$$C_a = W_h H_c + W_l L_c + W_f F_c \tag{18}$$

where, W_h, W_l and W_f are the weights of historical, local and future context respectively. These weights can map the different context representations with different dimensions into the same attention space.

2.6. Attention Mechanism

Attention networks are now a standard part of the deep learning toolkit, contributing to impressive results in neural machine translation (NMT) [12], image captioning, speech recognition etc. In case of sentiment analysis, not all the words are equally important to understand what the sentence means. Attention mechanism genuinely teaches the network on which words the network should focus [20] as it reduces the size of context representation.

$$\widetilde{h_t} = \tan(W_c[c_t : h_t]) \tag{19}$$

$$y_t = softmax(W_s\widetilde{h_t} + b_s) \tag{20}$$

Equations (19) and (20) are applied to the attention mechanism layer to compute the predictive probability of the words that matter, where h_t is the output of the first layer of the hidden state at each time step t. The goal is then to derive a context vector c_t that captures relevant source-side information to help predict the current target word y_t. For the target hidden state $\widetilde{h_t}$ and the source-side context vector c_t, a simple concatenation layer is combined. Then the information from both vectors produce an attentional hidden state as the equations (19) and (20). The attention vector h_t is then passed through a softmax layer to produce the output.

2.7. Gated-CNN

Gated-CNN is the most recent implementation of CNN to achieve better performance than LSTM. Gating mechanisms control the path through which information flows in the network. In case of LSTM, input and forget gate allow information to flow uninterrupted through many timesteps. Information could vanish through the transformations if the gates are missing. Unlike LSTM, convolutional network do not suffer from vanishing gradient problem and does not require forget gate. The gated mechanism controls the flow of the output by selecting what information should be propagated to the next layer. So, gated CNN will provide better accuracy than LSTM in case of sentiment analysis because the gated CNN will model the sentence to understand its meaning. The system works in two layers: convolutional layer and gated layer (Figure 5). The Lookup table in the figure is the context representation of the sentence. The context representation will be fed to the convolutional layer known as hidden layer and perform the computing of the hidden layer according to the equations.

$$h_t(X) = (X * W + b) \otimes \sigma(X * V + c) \tag{21}$$

The output of each layer is a linear projection $(X * W + b)$ modulated by the gates $\sigma(X * V + c)$. The output of convolution layer is multiplied with gates where the former output exists. This gating mechanism is known as Gated Linear Units (GLU). The convolution and gated linear unit in a pre-activation block that adds the input of the block to the output.

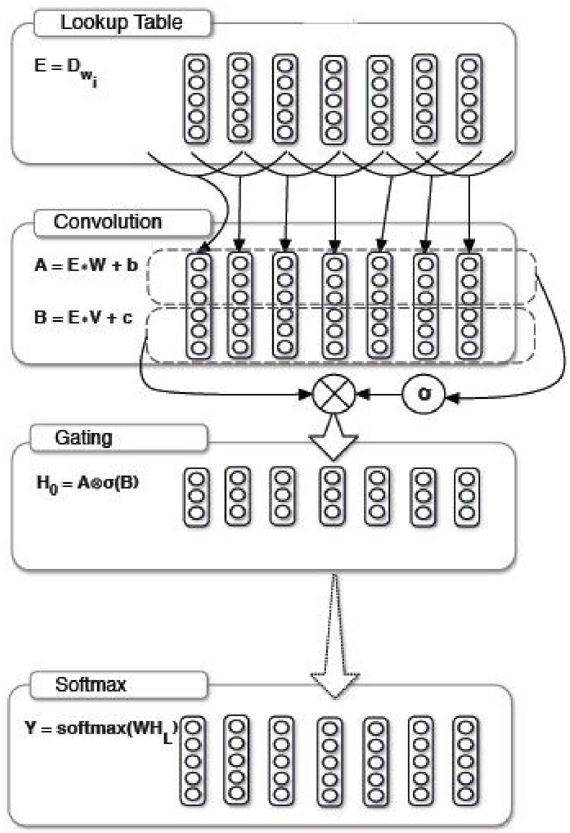

Figure 5: Architecture of Gated-CNN

3. Related Works

The classification of sentences on sentiment consists of several steps, namely, data cleaning, feature selection and classification process. Deep learning approaches [16] and vector representation of words have added new means for sentiment classification. One of the most impressive works done with neural networks is Richard Socher's work [15], centered on recursive neural networks.

Recurrent neural network (RNN) has already been successfully applied to text generation tasks [6], machine translation [21] as it can capture the relation between words and semantics of text. Another deep learning model is convolutional neural network (CNN) which is known for visual imagery analysis also works well on sentence classification [4] and sequence modeling [22].

Comprehensive attention recurrent models [5] performed efficiently in case of sentiment classification. Attention mechanism has shown an effective role in machine translation, sequence modeling [20], and sentiment analysis [13]. Gated-CNN is effectively used in language modeling [11]. It was able to resolve gradient related problems better than existing methods. Gated-CNN is used in sentiment analysis, semantic segmentation in recent times.

4. Classification Methodology

Our proposed systems are the moderation of comprehensive attention neural network model where we added or replaced layers for experimental purpose.

4.1. Implementation of Attention Mechanism with LSTM

Our first proposed model is designed with the addition of a new layer of attention mechanism with the comprehensive attention neural network model. The proposed model is shown in figure 6.

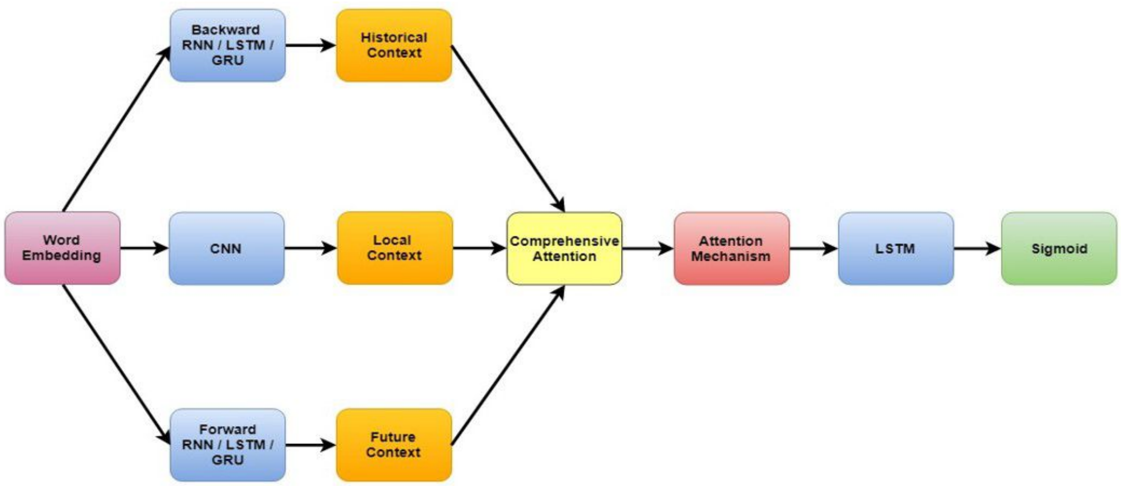

Figure 6: Proposed model with attention mechanism

The entire learning algorithm of attention mechanism with comprehensive attention recurrent model is summarized as Algorithm 1.

Algorithm 1	Pseudo-code for Attention Mechanism with Comprehensive Attention Recurrent Models
1	Construct word embedding using word2vec from the input dataset.
2	Implement backward RNN/LSTM/GRU to obtain the historical context representation from equation (15) using equations (1), (9) and (13).
3	Implement CNN to obtain the local context representation from equation (16) using equation (14).
4	Implement forward RNN/LSTM/GRU to obtain the future context representation from equation (17) using equations (2), (9) and (13).
5	Combine the historical, local and future context representation with weighted sum method to obtain the comprehensive attention context representation using equation (18).
6	Feed the context representation to the attention mechanism layer to reduce the vector length which is performed using equation (19) and (20).
7	Pass the reduced vector representation to the LSTM to obtain the sentence representation.
8	Apply sigmoid function to produce the conditional probability over the class labels.

4.2. Implementation of Gated-CNN

Our second model consists of a Gated-CNN network in replacement of the long-short-term-memory (LSTM) layer. As discussed previously, the implementation of comprehensive attention recurrent neural model remains the same except the comprehensive context representation is passed through a gated convolutional neural network instead of an LSTM network.

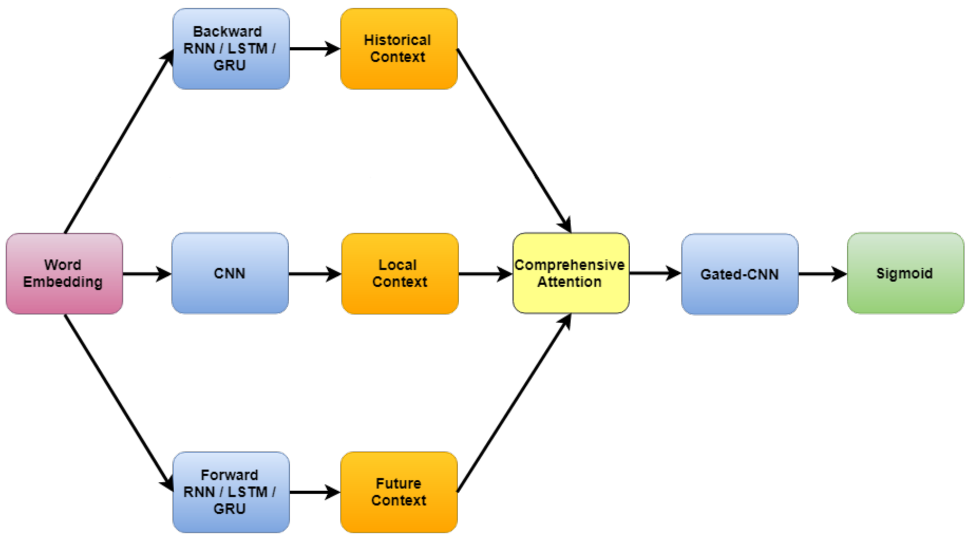

Figure 7: Proposed model with Gated-CNN

The entire learning algorithm of attention mechanism with comprehensive attention recurrent model is summarized as Algorithm 2.

Algorithm 2	Pseudo-code for Comprehensive Attention Recurrent Models with gated-CNN
1	Construct word embedding using word2vec from the input dataset.
2	Implement backward RNN/LSTM/GRU to obtain the historical context representation from equation (15) using equations (1), (9) and (13).
3	Implement CNN to obtain the local context representation from equation (16) using equation (14).
4	Implement forward RNN/LSTM/GRU to obtain the future context representation from equation (17) using equations (2), (9) and (13).
5	Combine the historical, local and future context representation with weighted sum method to obtain the comprehensive attention context representation using equation (18).
6	Pass the reduced vector representation to gated-CNN to obtain the sentence representation using equation (21).
7	Apply sigmoid function to produce the conditional probability over the class labels.

4.3. Implementation of Attention Mechanism with Gated-CNN

Finally, we have implemented attention mechanism with gated-CNN to reduce the loss function and increase the accuracy as attention layer increases efficiency.

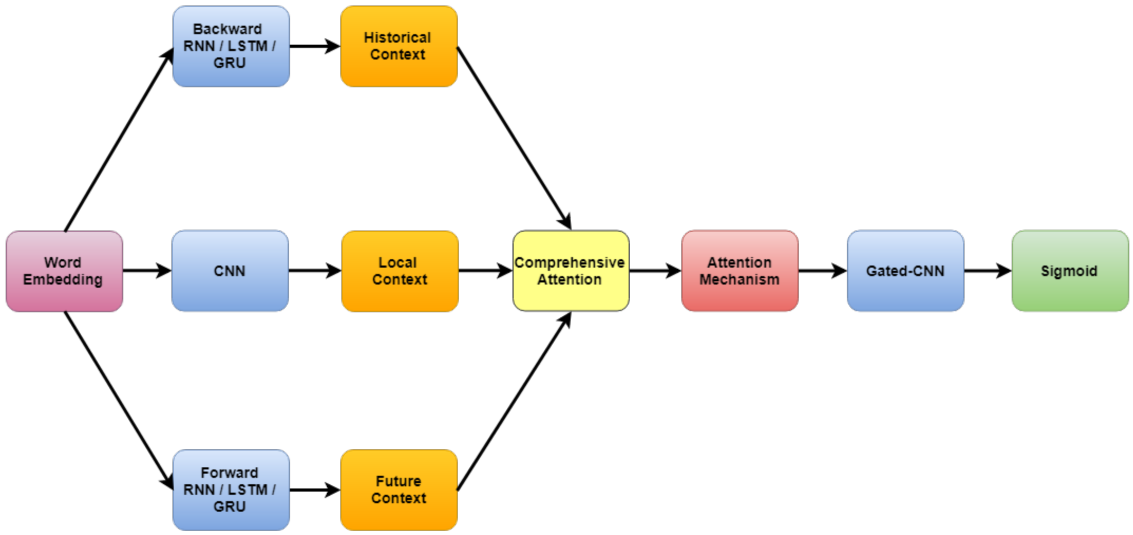

Figure 8: Proposed model with attention mechanism along with Gated-CNN

The model given below performs better than the state-of-the-arts method in case of sentiment classification. The entire learning algorithm of attention mechanism with comprehensive attention recurrent model is summarized as Algorithm 3.

Algorithm 3	Pseudo-code for Attention based gated-CNN with Comprehensive Attention Recurrent Models
1	Construct word embedding using word2vec from the input dataset.
2	Implement backward RNN/LSTM/GRU to obtain the historical context representation from equation (15) using equations (1), (9) and (13).
3	Implement CNN to obtain the local context representation from equation (16) using equation (14).
4	Implement forward RNN/LSTM/GRU to obtain the future context representation from equation (17) using equations (2), (9) and (13).
5	Combine the historical, local and future context representation with weighted sum method to obtain the comprehensive attention context representation using equation (18).
6	Feed the context representation to the attention mechanism layer to reduce the vector length which is performed using equation (19) and (20).
7	Pass the reduced vector representation to gated-CNN to obtain the sentence representation using equation (21).
8	Apply sigmoid function to produce the conditional probability over the class labels.

5. Experimental Results and Discussions

5.1. Dataset

We evaluated the performance of our proposed models on IMDB dataset [14]. This dataset has 50,000 movie reviews with 1:1 binary labeling. We split the dataset randomly 50:50 into training and testing sets. Reviews have been pre-processed, and each review is encoded as a sequence of word indexes (integers). We have considered the top words (highest number of texts taken from each movie review) as 10,000, max movie review length is 1600 and embedding vector size is 300.

The number of hidden layer nodes of recurrent layers, convolutional layers, attention layers and gated CNN layers vary as per the models to get the best out of it. The training batch size is set to 128. All the implementations of the models are conducted using a GeForce 840M GPU on a windows PC with 2.2 GHZ CPU and 8 GB of RAM.

5.2. Experimental Results

The performance of our model is compared with several baseline methods. After training, each model is tested with test reviews around 10 times and averaged the results. The average accuracy shows the percentage of correct classification of sentiment. The average precision and recall is calculated to understand the effectiveness of our model. In order to understand the performance of the proposed models as a whole, f1-score is calculated.

TABLE 1: Classification Accuracy (%), Precision (%), Recall (%) and F1-score (%) comparison results of models using recursive neural network (RNN)

Methods	Accuracy	Precision	Recall	F1-scores
RNN [15]	83.13	87.51	87.82	87.79
CA-RNN [5]	89.00	89.44	89.32	89.38
CA-RNN with attention	**89.33**	**90.10**	**89.41**	**89.30**
CA-RNN with gated-CNN	**89.50**	**89.71**	**89.55**	**89.37**
CA-RNN with attention based gated-CNN	**89.07**	**91.27**	**88.89**	**89.61**

The comparison results given in TABLE 1 illustrate information about attention mechanism, gated-CNN and attention mechanism with gated-CNN using recurrent neural network (RNN). The comparison is made based on the result taken from the original paper [5]. The results shown in TABLE 1 shows an impressive performance as a whole. But the model CA-RNN with gated-CNN produces a better result than the model with attention mechanism and the merged version of attention mechanism with gated-CNN. This is because in case of recursive network, the review words cycle through the nodes and gated-CNN controls the flow of the words which is significant for classification. Though CA-RNN with gated-CNN has the best accuracy, the model of CA-RNN with attention based gated-CNN shows a better precision and recall, proving it as a better performing model than the others.

TABLE 2: Classification Accuracy (%), Precision (%), Recall (%) and F1-score (%) comparison results of models using long short term memory (LSTM)

Methods	Accuracy	Precision	Recall	F1-scores
LSTM [23]	85.34	87.88	85.21	88.22
CA-LSTM [5]	90.10	90.14	90.04	90.09
CA-LSTM with attention	**90.44**	**90.89**	**89.51**	**89.20**
CA-LSTM with gated-CNN	**90.11**	**88.73**	**90.19**	**88.05**
CA-LSTM with attention based gated-CNN	**91.02**	**91.53**	**89.77**	**90.03**

The comparison Table 2 demonstrates the results of proposed models with LSTM. However, our modified method CA-LSTM with attention is very precise and accurate. Also with the help of gated-CNN the model still shows a better result than its basic form. According to accuracy measures, CA-LSTM with attention based gated –CNN beats all other models of similar architecture. The model CA-LSTM with attention based gated-CNN shows a great performance overall among all four categories and achieves a very good result.

TABLE 3 : Classification Accuracy(%), Precision(%), Recall(%) and F1-score(%) comparison results of models using gated recurrent unit (GRU)

Methods	**Accuracy**	**Precision**	**Recall**	**F1-scores**
GRU [24]	86.00	84.55	87.33	88.10
CA-GRU [5]	90.10	90.24	89.97	90.11
CA-GRU with attention	**90.21**	**90.36**	**89.70**	**89.90**
CA-GRU with gated-CNN	**90.67**	**93.79**	**88.09**	**90.34**
CA-GRU with attention based gated-CNN	**91.44**	**92.41**	**89.88**	**91.10**

The comparison given in TABLE 3 provides the results of our models with gated recurrent unit (GRU) which produce the best outcome among all the three proposed models. Though there is a slight increase in accuracy with CA-GRU with attention, the other two models CA-GRU with gated-CNN and CA-GRU with attention based gated-CNN also show promising accuracy. The precision and recall of these two models also indicate that their performances are pretty good too. The model CA-GRU with attention based gated-CNN provides the best performing result among all the proposed models we have implemented so far.

In general, the models we proposed have really performed well in all aspects. The basic model with comprehensive attention performs well as the models focus on comprehensive context representation. When attention layer is added, it updates the weight matrix for better learning and the performance improved consequently. Replacement of LSTM in the model with gated-CNN decreases the gradient based problem. When comprehensive attention is added with GRU it enhances the effectiveness of the model as it reduces the complexity. Merging the gated-CNN with CA-GRU is supposed to increase the performance as we have already noticed so far.

5.3. Precision-Recall Curve

Precision-Recall curve (PR-curve) is a medium of visualizing performance of various algorithms [18]. Recall is the ratio of correctly predicted positive observations to the all observations in actual class, whereas precision is the ratio of correctly predicted positive observations to the total predicted positive observations. Any model that has an average precision of 0.7-0.9 are considered to be well-performed. The straightness of the curve reflects stability of models.

Figure 9 illustrates the precision and recall of the models using CA-RNN. The models show average precision recall score (AP) of 0.96 for CA-RNN with attention and AP of 0.95 for other two models with CA-RNN. The precision of CA-RNN with gated-CNN is comparatively less than the other models. Each model shows best possible outcome for sentiment analysis.

For models using CA-LSTM, the model CA-LSTM with attention mechanism with gated-CNN shows the most

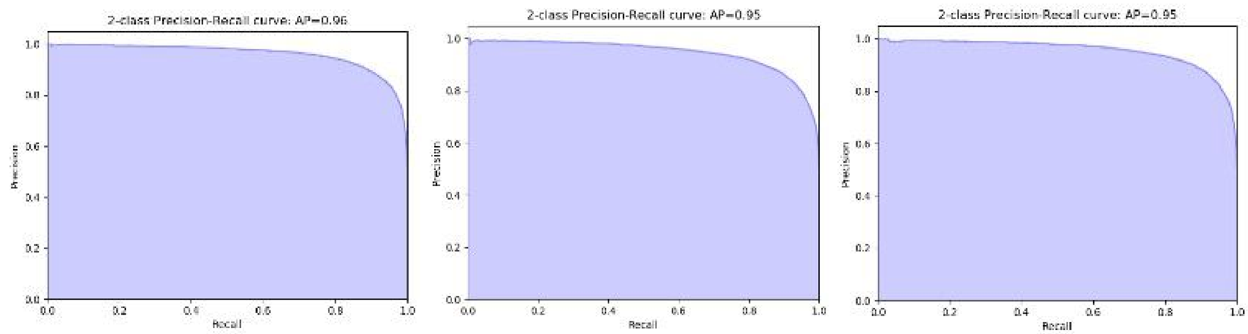

Figure 9: Precision-recall curve for models of CA-RNN with attention mechanism, gated-CNN and attention mechanism with gated-CNN respectively

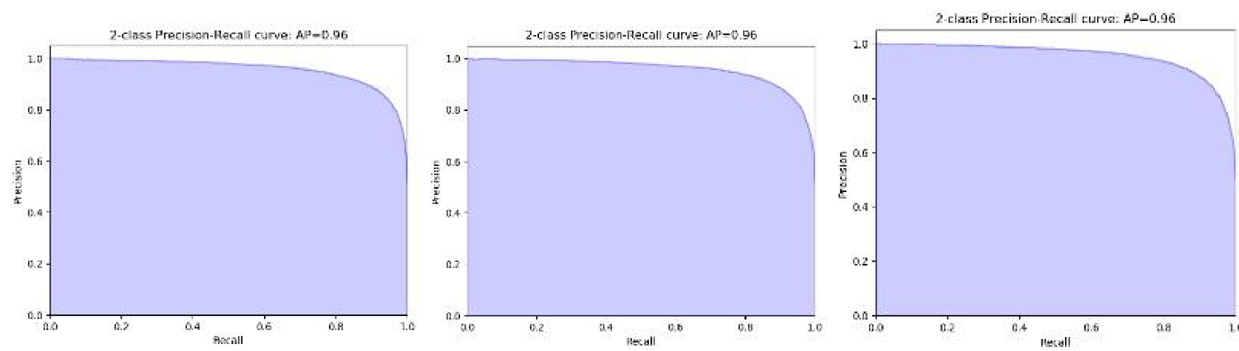

Figure 10: Precision-recall curve for models of CA-LSTM with attention mechanism, gated-CNN and attention mechanism with gated-CNN respectively

precise results in figure 9 compared with others. The models show average precision recall score (AP) of 0.96. Though the PR curves have curved features, yet these curves are quite straight in nature, making the positivity prediction rate next to perfection. Then again with a slight difference CA-LSTM with attention mechanism with gated-CNN can be considered as the best model among all.

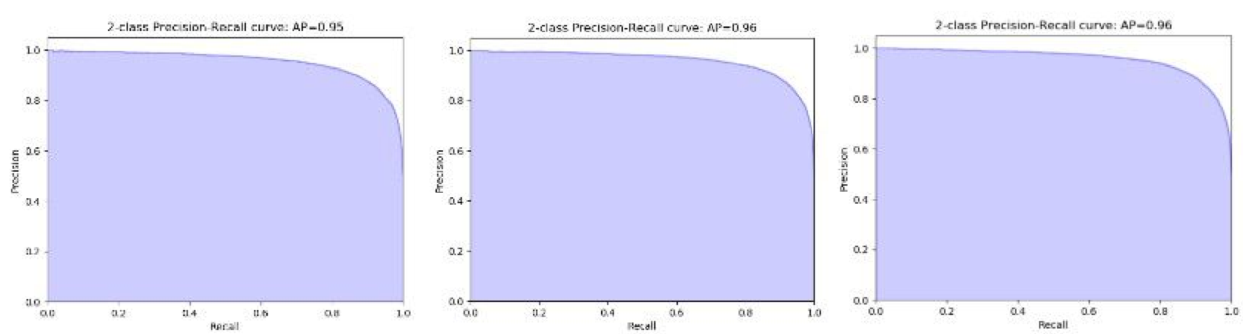

Figure 11: Precision-recall curve for models of CA-GRU with attention mechanism, gated-CNN and attention mechanism with gated-CNN respectively

Similarly figure 11 demonstrates that the CA-GRU with attention mechanism is poorer than the models with gated-CNN and attention mechanism with gated-CNN. In case of CA-GRU, the method with gated-CNN is slightly better than the other two.

Our models show an average precision rate of 0.95-0.96. All the PR-curves presented in Figure 9, Figure 10 and Figure 11 have much straighter line, which indicate the stability of the performance. The results from TABLE 1, TABLE 2 and TABLE 3 and from PR-curve suggest that, the models enhances the accuracy by reducing the learning complexity. All the models are precise and well trained as a whole and can be applied on several datasets.

6. Conclusions

In this paper, our main objective was to enhance the effectiveness of the model [5] by adding attention layer and making a few changes. In almost every case, proposed models have been able to perform better than the current model [5]. The attention layer has been used to reduce the complexity of the model by making training less time consuming. The comprehensive attention model with gated-CNN successfully performs better than the one with LSTM. The proposed model with both attention mechanism and gated-CNN proves to be very promising as it enhances the performance of the classifier. The experimental results on the IMDB dataset which is trained end-to-end, show that the proposed model has been able to perform well in terms of both accuracy and precision. In future, we will use some single hidden layer feedforward networks [17] to make our model more time efficient. There have been some modifications made in latest version of gated-CNN, we will incorporate with that to make our model more precise and accurate.

Acknowledgment

The research was performed as a thesis topic to pursue Bachelor degree in Computer Science & Engineering from Khulna University, Khulna, Bangladesh. It was done under the supervision of Professor S M Rafizul Haque. S. M. Mushfiqur Rahman and Ashmita Riya found the idea and implemented the code.

References

[1] Das MK, Padhy B, Mishra BK. Opinion mining and sentiment classification: A review. 2017 International Conference on Inventive Systems and Control (ICISC), Coimbatore, 2017, 1-3. Doi: 10.1109/ICISC.2017.8068637

[2] Pang B, Lee L and Vaithyanathan S. Thumbs up? Sentiment Classification using Machine Learning Techniques. Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing-Volume 10. Association for Computational Linguistics, 2002, 79–86. Doi: 10.3115/1118693.1118704

[3] Mullen T and Collier N. Sentiment Analysis using Support Vector Machines with Diverse Information Sources. Proceedings of the 2004 conference on empirical methods in natural language processing, 2004, 412–418

[4] Kim Y. Convolutional Neural Networks for Sentence Classification. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing Association for Computational Linguistics, 2014. Doi: 10.3115/v1/D14-1181

[5] Zhang Y, Meng JE, Venkatesan R, Wang N, Pratama M. Sentiment classification using Comprehensive Attention Recurrent models. 2016 International Joint Conference on Neural Networks (IJCNN), 2016. Doi:10.1109/ijcnn.2016.7727384

[6] Sutskever I, Martens J, Hinton GE. Generating Text with Recurrent Neural Networks. Proceedings of the 28th International Conference on Machine Learning (ICML-11), 2011, 1017-1024

[7] Schuster M, Paliwal KK. Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing, 1997, 45(11):2673-2681. Doi: 10.1109/78.650093

[8] Johnson R and Zhang T. Supervised and Semi-supervised Text Categorization using LSTM for Region Embedding. International Conference on Machine Learning, New York, USA, 2016.

[9] Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F et al. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation, 2014. Doi:10.3115/v1/D14-1179

[10] Yin W, Kann K, Yu M, Schütze H. Comparative Study of CNN and RNN for Natural Language Processing, 2017. arXiv:1702.01923

[11] Dauphin YN, Fan A, Auli M, Grangier D. Language Modeling with gated Convolutional Networks. 34th International Conference on Machine Learning, Sydney, Australia, 2017. 1-20. arXiv:1612.08083v3

[12] Luong MT, Pham H, Manning CD. Effective approaches to Attention Based Neural Machine Transformation, 2015. arXiv: 1504025 V5 [cs.CL]

[13] Wang Y, Huang M, Zhu X, Zhao L. Attention-based LSTM for Aspect level Sentiment Analysis. Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 2016, 606–615. Doi:10.18653/v1/D16-1058

[14] Maas A, Daly RE, Pham PT, Huang D, Ng AY, Potts C. Learning word vectors for sentiment analysis. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, 2011, 1:142-150.

[15] Socher R, Perelygin A, Wu J, Chuang J, Manning CD, Ng AY, Potts C. Recursive deep models for semantic compositionality over a sentiment treebank. Proceedings of the conference on empirical methods in natural language processing (EMNLP), 2013, 1631: 1642.

[16] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature 2015; 521: 436-444. https://doi.org/10.1038/nature14539

[17] Wang N, Er M J, Han M. Generalized single-hidden layer feedforward networks for regression problems. Neural Networks and Learning Systems, IEEE Transactions, 2015 , 26(6): 1161-1176. Doi: 10.1109/TNNLS.2014.2334366

[18] Davis J, Goadrich M. The Relationship Between Precision-Recall and ROC Curves. Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, PA, 2006. Doi: 10.1145/1143844.1143874

[19] Mikolov T, Chen K, Corrado G, Dean J. Efficient Estimation of Word Representations in Vector Space. 2013, 1-12. arXiv: 1301.3781.

[20] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L et al. Attention Is All You Need. 31st Conference on Neural Information Processing Systems (NIPS), USA, 2017.

[21] Auli M, Galley M, Quirk C, Zweig G. Joint language and translation modeling with recurrent neural networks. Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, 2013, 3(8):1044–1054.

[22] Kalchbrenner N, Grefenstette E, Blunsom P. A convolutional neural network for modelling sentences. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2014. Doi: 10.3115/v1/P14-1062

[23] Li D, Qian J. Text sentiment analysis based on long short-term memory. 2016 First IEEE International Conference on Computer Communication and the Internet (ICCCI), Wuhan, 2016, 471-475. Doi: 10.1109/CCI.2016.7778967

[24] Hu F, Li L, Zhang ZL, Wang JY, Xu XF. Emphasizing essential words for sentiment classification based on recurrent neural networks. Journal of Computer Science and Technology. 2017, 785–795. Doi:10.1007/s11390-017-1759-2