International Journal of Machine Learning and Cybernetics (2020) 11:2637–2651
https://doi.org/10.1007/s13042-020-01135-1

ORIGINAL ARTICLE

Attentive convolutional gated recurrent network: a contextual model to sentiment analysis

Olivier Habimana[1] · Yuhua Li[1] · Ruixuan Li[1] · Xiwu Gu[1] · Wenjin Yan[1]

Received: 20 April 2019 / Accepted: 2 May 2020 / Published online: 8 June 2020
© Springer-Verlag GmbH Germany, part of Springer Nature 2020

Abstract
Considering contextual features is a key issue in sentiment analysis. Existing approaches including convolutional neural networks (CNNs) and recurrent neural networks (RNNs) lack the ability to account and prioritize informative contextual features that are necessary for better sentiment interpretation. CNNs present limited capability since they are required to be very deep, which can lead to the gradient vanishing whereas, RNNs fail because they sequentially process input sequences. Furthermore, the two approaches treat all words equally. In this paper, we suggest a novel approach named attentive convolutional gated recurrent network (ACGRN) that alleviates the above issues for sentiment analysis. The motivation behind ACGRN is to avoid the vanishing gradient caused by deep CNN via applying a shallow-and-wide CNN that learns local contextual features. Afterwards, to solve the problem caused by the sequential structure of RNN and prioritizing informative contextual information, we use a novel prior knowledge attention based bidirectional gated recurrent unit (ATBiGRU). Prior knowledge ATBiGRU captures global contextual features with a strong focus on the previous hidden states that carry more valuable information to the current time step. The experimental results show that ACGRN significantly outperforms the baseline models over six small and large real-world datasets for the sentiment classification task.

Keywords Sentiment analysis · Convolutional neural network · Recurrent neural network · Attention mechanism · Contextual features

1 Introduction

Nowadays, with the notable increase of Web 2.0 tools like online social media and e-commerce platforms, users freely express their ideas and thoughts in the form of text [1, 5, 6, 39, 58]. Consequently, many organizations became increasingly interested in getting the hidden insights from these user-generated content (UGC) [13, 24, 36] to assist in decision making and monitoring public opinion. Therefore, sentiment analysis has received a substantial amount of attention from many researchers as one of the natural language processing tasks that focuses on finding the opinions articulated in the UGC.

To get good results in sentiment analysis requires modeling and prioritizing informative contextual features. Considering the following review text extracted from the Amazon dataset, which talks about the sandals: "I received this day and I'm not a fan of it but I thought it would be puffier as it looks in the pic but it is not what I wanted to do with the sandals she was gonna wear it now I'm going to find another pair of sandals, just keep it cuz she likes it". In view of local features like word-based features in the short sub sentences "i am not a fan" and "it is not what I wanted", one may judge the review for being negative while at the end, the sub sentence "she likes it" is positive. However, it is difficult to find the sentiment polarity of this review without considering the usage of both local and global contextual features and ignoring irrelevant words, since they introduce noise.

In the present work, to get the sentiment label of review text like in such above example, we focus on the extraction of two main types of contextual features. The first type is local contextual features like n-grams and negation, which highly depend on the order of words in a text sequence [34]. The order matters because the contextual polarity of

✉ Yuhua Li
idcliyuhua@hust.edu.cn

✉ Ruixuan Li
rxli@hust.edu.cn

[1] School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China

given words can change depending on their position in the text sequence [49]. To capture these features, a subset of the input sequence must be considered. The second type is global contextual features that are reliant on long-range dependencies in the text sequence. We extract global long-range dependencies to bring the relevant contexts closer to the words being predicted. Thus, to deal with these global contextual features requires having the whole context of the text sequence [28, 46].

Recently, deep learning methods such as convolutional neural networks (CNNs) [9] and recurrent neural networks (RNNs) like long short-term memory (LSTM) [14] and gated recurrent unit (GRU) [8] have considerably improved the results in sentiment analysis without laborious feature engineering. CNNs based approaches have demonstrated the capability of modeling local contextual features because they are unbiased models. Besides, they preserve the spatial structure of the input sentence [16]. RNNs, on the other hand, have recently shown promising results in learning global long-range dependencies because they can maintain the constant error flow [35]. To enhance the capability of RNNs in dealing with global long-range dependencies and capturing salient features in the input sequence, researchers have proposed the use of the attention mechanism [3, 28, 40]. RNN attention-based models present encouraging results on different tasks of sentiment analysis [18].

Although these models have achieved impressive results, their behavior for modeling contextual features is still unsatisfactory. First, CNNs fail to capture global contextual features because they need to be very deep [10, 17]. It is, therefore, easy for the gradient to vanish during the training process. Second, RNNs are limited to model local contextual features because they lack the task-specific structure [10]. Lastly, the capacity of RNNs for modeling global contextual features is constrained to the fact that RNNs process input sequences in sequential order, where the current time step depends on one previous hidden state. As a result, it implies RNNs to have the bias of favoring recent inputs, which causes them to be limited to a certain extent while modeling long-range dependencies [48]. Even though attention-based models were proposed to help RNN in dealing with long-range dependencies and learning relevant features in the input text sequence, we argue that they are still generic models to deal with both local and global features. This argument is linked to the fact that in this context, the attention mechanism is computed to find the representation of the whole input sequence when the last RNN unit outputs, i.e., is applied on the output sequence of sequential RNNs. Thus, RNN attention-based models suffer the same problem as original RNNs, which is linked to the sequential processing of the input text sequence.

To address these problems, we propose a novel model named attentive convolutional gated recurrent network (ACGRN) that uses local and global contextual features to make the final prediction of the input sequence. ACGRN model is distinctive in that it uses a shallow-and-wide CNN to extract extreme local features. Then, it applies our novel prior knowledge attention-based bidirectional GRU (ATBi-GRU) that allows learning global contextual features by feeding the contextual information of all prior hidden states (prior knowledge) to the current time step. Different from previous attention-based models, in our model, the attention is computed at each time step to give the current time step the prior knowledge of all past hidden states. In summary, the contributions of this paper are as follows:

1. We propose a shallow multichannel CNN followed by a max-pooling layer to learn local contextual features and produce high-level representations.
2. A novel prior knowledge attention based bidirectional GRU (ATBiGRU) is brought up to extract global contextual features. Specifically, it allows the current time step to have access to the aggregated representation of all previous hidden states.
3. We evaluate the performance of our ACGRN model on six real-world datasets. The proposed ACGRN model outperforms state-of-the-art approaches in terms of accuracy. Some visualization cases also validate the effectiveness of ACGRN model.

The rest of the paper is structured as follows. First, we discuss several related works in Sect. 2. Second, we give a detailed description of ACGRN architecture in Sect. 3. Third, the experimental setup and results achieved by our model are discussed in Sect. 4. In Sect. 5, we provide the discussion and qualitative analysis of our model. Finally, in Sect. 6 we conclude the paper with a final remark.

2 Related work

In this section, we briefly discuss deep learning methods for sentiment analysis aiming to model local and global contextual features. These approaches fall into three categories: CNN-based, RNN-based, and hybrid approaches.

2.1 CNN-based approaches

CNN-based models have shown superior performance in modeling local contextual features using filters. A two-channel CNN-based approach was explored to extract a possible number of local contextual features [20]. The two channels receive different inputs, where the first one is treated as static while the second is fine-turned during the training process. The study [56] conducted a sensitivity analysis of a one-layer CNN model to prove the effectiveness of its different

components on the performance. Similarly, shallow-and-wide CNN, as well as DenseNet, were explored to evaluate the effect of CNN's deepness on the performance [23]. The study proved that shallow-wide CNN is most effective at word-level compared to the deep CNN. A CNN-based approach that takes into account the word order in extracting local contextual features for text classification has been explored [16]. A dynamic CNN-based model with a dynamic k-max-pooling mechanism was proposed to handle short and long-range dependencies in the input sentence [19]. With this dynamic CNN-based model, the convolutional layers alternate with the max-pooling layer, where k-max values are pooled in the input sentence. A two-layer deep CNN was designed to exploit character-to-sentence-level features in the input sentences of any size [12]. Shallow CNN-based approaches can only deal with local contextual features bounded in their window sizes.

Consequently, researchers suggested very deep CNN-based models for capturing global long-range contextual features. A very deep CNN with 29 small convolutional layers that operate at the character level was proposed to capture long-range association in a sentence [10]. It proved that the performance of CNN increases with depth. Similarly, the study [17] introduced a deep pyramid CNN to deal with long-range dependencies. The proposed CNN model tried to improve the performance of word-level CNN by increasing its depth in a deep pyramidal shape where its internal structure is reduced over time.

Although CNN-based approaches have shown promising capability in extracting local contextual features, they are limited to learn global contextual features as they are required to be very deep. Besides, the local contextual features extracted by CNN are not satisfactory for sentiment interpretation.

2.2 RNN-based approaches

In the literature, a large number of sequential RNN-based models have been suggested to learn long-range contextual features in general. A generative contextual bidirectional-LSTM (cBiLSTM) model was introduced to predict the sentiment label of a word based on its right and left contexts in the sentence [33]. Likewise, the study [54] proposed a neural network that uses a bi-directional gated recurrent network (BiGRN) to link together the input tweets. Then, it applies a three-way gated network model to the produced hidden states to learn the relationship between the target word and its surrounding contexts. In order to deal with long-term dependencies, a BiLSTM was suggested to capture the contextual information from the input texts represented by considering the weight of each word [51]. The work [43] constructed a tree-LSTM model that represents the words of the sentence in the form of the parent and child relationship. The proposed tree-LSTM approach presented the capability to deal with long-range dependencies. Researchers in [44] came up with a capsule tree-LSTM approach for addressing the bias limitation of LSTM as well as tree-LSTM. The model introduces a dynamic routing algorithm as an aggregation layer, which is used in the sentence representation construction to automatically learn the weights of each node.

On the other hand, the recent development in research has brought a new idea of using attention mechanism [3, 28] to allow the model to focus on the most contributing part of the input sequence. Correspondingly, different attention-based approaches have been explored in sentiment analysis. A hierarchical GRU attention-based network was suggested to hierarchically learn informative information of the words and sentences of a document [53]. The work [52] suggested a bidirectional LSTM with the attention mechanism to model the relationship between the target word and its discriminative features in the input sentence. Researchers [45] designed a BiGRU coupled with attention mechanism for capturing the long-range dependencies in the input sentence. Likewise, the authors in [38] introduced an attention-based tree structure GRU model that represents the sentence by integrating the structure information at each node of the tree. In this approach, they used the attention mechanism to prioritize the most contributing nodes in the tree. Similarly, the work [50] designed a model that takes into consideration user and product information in sentiment classification. The proposed model separately uses the bidirectional LSTM, followed by an attention mechanism to generate the user and the product representations, which are combined to make the final representation. A cognitive-based attention LSTM approach that uses the attention mechanism, which is built using the cognition ground eye-tracking data was introduced to deal with long-term dependencies in sentiment analysis [27]. A hierarchical LSTM-based model trained by cognition grounded eye-tracking data that predicts overall review text's sentiment was suggested [32]. A BiLSTM with multi-head attention was proposed to deal with the long-term dependency problem as well as to capture the actual context of the text [26].

However, despite the success of RNN-based approaches, they are limited to deal with local contextual features since they are biased to favor the recent inputs. Furthermore, they are bound to extract global contextual features due to the recurrent nature.

2.3 Hybrid neural networks

Currently, there is a large body of hybrid models in which researchers attempt to combine the advantages of CNN and RNN by using them to extract local and global contextual features. The work [11] constructed a hybrid model that augments the CNN with LSTM. The pooling layer

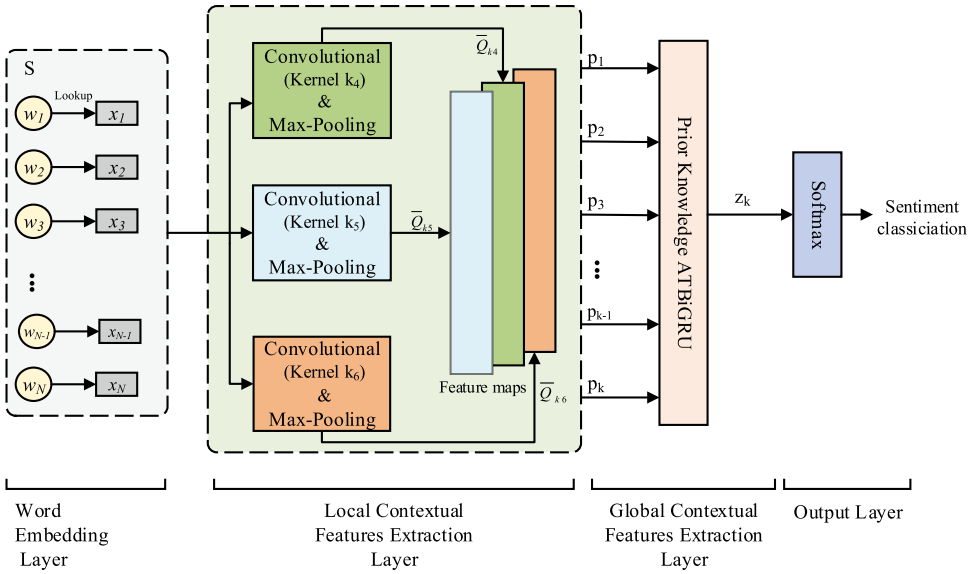

Fig. 1 The overall architecture of attentive convolutional gated recurrent network (ACGRN)

of the proposed CNN is replaced with an LSTM layer for better capturing long-range dependencies. Researchers in [55] designed a dependency sensitivity CNN model that learns long-term dependencies and generates a hierarchical representation of a sentence using an LSTM, and then applies the convolutions to learn local contextual features. The work [22] proposed a model that uses the max-pooling layer to select relevant features among the contextual features produced by bidirectional RNN. A hybrid contextualized sentiment classifier model, which combines a CNN model applied to learn local feature sentences and a BiLSTM that deals with long-term dependencies, was proposed [2]. A self-attention sandwich neural network was suggested to learn local semantic and global structure representations [57]. The proposed model makes full use of both representations with a self-attention mechanism. The study [29] introduced a framework with a mutual attention mechanism to exploit the mutual effects between local contextual features extracted with CNN and global contextual features learned with BiLSTM.

Despite the success of these models, they are generic appraoches to extract contextual features. Therefore, in this paper, we focus on designing a simple multi-channel CNN that allows extracting local contextual and a prior knowledge attention-based bidirectional GRU that allows the current time step to have access to the aggregated representation of all previous hidden states. Precisely, the previous hidden states that carry more valuable information to the present time step are prioritized by the attention mechanism.

3 Proposed method

In this section, we discuss the details of our proposed model to deal with contextual features in sentiment analysis. We first describe the task definition, followed by a shallow overview of our model architecture. Lastly, we describe our model architecture layer-by-layer.

3.1 Task definition

We argue that in order to obtain good results in sentiment analysis, it is necessary to consider local and global contextual features. Thus, in this work, we give the following analogy for extracting these contextual features. Considering an input text sequence S with length N, $S = [x_1, x_2, x_3, x_4, \ldots, x_N] \in \mathbb{R}^{d \times N}$ where $x_i \in \mathbb{R}^d$ corresponds to the ith word vector in the text sequence matrix. We aim to assign to the text sequence S a sentiment label. We assert that to find the polarity of a given input text sequence S, each word x_i in the text sequence S holds the key local and global contextual features necessary for sentiment interpretation. Thus, exploiting the complement of these contextual features can help to improve the performance.

3.2 Model overview

To address the above-described problem, we propose the ACGRN model shown in Fig. 1. It consists of two main components: the local contextual features extraction layer and the global contextual features layer. The former uses

convolutional and max-pooling layers to extract local contextual features and generates high-level representation from embeddings. It applies three concurrent convolution operations with kernel window sizes k_4, k_5, and k_6. Afterwards, the pooling layer is used to downscale the features by extracting the maximum between two contiguous features in the feature map. The resulted high-level representations are concatenated and fed to the latter component that uses prior knowledge attention based BiGRU to extract global contextual features. Specifically, this component allows the current time step to have access to all previous hidden states. Finally, the model applies a softmax classifier to generate the prediction based on the extracted contextual features.

3.3 Word embedding layer

The input to the proposed model is a text sequence S of length N denoted as $S = [w_1, w_2, w_3, \ldots, w_{N-1}, w_N]$. The embedding layer maps each word in the input sequence S to a high-dimensional vector space through the pre-trained GloVe embedding method [37]. Accordingly, the output of this layer is an embedding matrix $S = [x_1, x_2, x_3, \ldots, x_{N-1}, x_N] \in \mathbb{R}^{N \times d}$, where $x_i \in \mathbb{R}^d$ corresponds to the ith word vector in the sequence representation S and d is the embedding dimension.

3.4 Local contextual features extraction layer

Inspired by the performance of CNN based-models described in the literature, we apply three concurrent shallow convolutional layers followed by max-pooling layers to extract local contextual features and generate a high-level representation.

In general, the structure of a convolutional layer, which is applied to the text sequence representations depends on the length of the text sequence and embedding dimension denoted by N and d, respectively. The convolutional layer applies a filter with weight matrix $F \in \mathbb{R}^{n \times d}$ to each possible window of n words of the sequence matrix S and generates a feature map M. Formally, the ith element of the feature map M generated from n-gram text fragment is defined as follows:

$$m_i = \sigma\Big(\sum (S[*, i : i+n] \odot F) + b\Big) \tag{1}$$

where $b \in \mathbb{R}$ is a bias term, and σ is a non-linear function, which can be either sigmoid, hyperbolic tangent, or rectified linear unit. $\odot$ is the Hadamard product between two matrices. Thus, the filter F is applied to each possible window of words in the text sequence matrix S, i.e., $x_{1:n}, x_{2:n+1}, \ldots, x_{N-n+1:N}$ to generate feature map $M \in \mathbb{R}^{N-n+1}$, which is expressed as follows:

$$M = [m_1, m_2, m_3, \ldots, m_{N-n+1}] \tag{2}$$

Motivated by the performance of pooling operation for dimension reduction and noise reduction, we adopt max-pooling rather than average pooling. Max-pooling retains extreme features with less computational complexity [9]. On the other hand, the average pooling may not extract informative features as it computes the average of all values, which may or may not be necessary. Accordingly, the max-pooling operation transforms the feature map M to $Q \in \mathbb{R}^{\lfloor \frac{N-n+1}{2} \rfloor}$, which is defined as follows:

$$Q = [q_1, q_2, q_3, \ldots, q_{\lfloor \frac{N-n+1}{2} \rfloor}] \tag{3}$$

where the ith element of the feature map q is expressed as follows:

$$q_i = \max(m_{2\times i-1}, m_{2\times i}) \tag{4}$$

To get sufficient local contextual features, we, therefore, apply L different filters to get L feature maps, which can be rearranged through column vector concatenation as follows:

$$\bar{Q} = [Q^1, Q^2, Q^3, \ldots, Q^L] \tag{5}$$

where $\bar{Q} \in \mathbb{R}^{\lfloor \frac{N-n+1}{2} \rfloor \times L}$

Furthermore, stimulated by Kim [20], we apply three convolutional channels with kernel window size of 4, 5, and 6 to get feature maps $\bar{Q}_{k4}, \bar{Q}_{k5}$, and $\bar{Q}_{k6}$, respectively. Thus, to get the final feature maps $P \in \mathbb{R}^{\lfloor \frac{N-n+1}{2} \rfloor \times (L \times 3)}$, we concatenate Q_{k4}, Q_{k5}, and Q_{k6} as follows:

$$P = [\bar{Q}_{k4}, \bar{Q}_{k5}, \bar{Q}_{k6}] \tag{6}$$

For simplicity, let k and $\bar{d}$ denote $\lfloor \frac{N-n+1}{2} \rfloor$ and $L \times 3$, respectively. Therefore, the matrix P with its elements is written as $P = [p_1, p_2, p_3, \ldots, p_{k-1}, p_k] \in \mathbb{R}^{k \times \bar{d}}$. Then, we feed these feature maps to a prior knowledge attention based BiGRU to capture global contextual features.

3.5 Global contextual features extraction layer

In this subsection, we first introduce the standard sequential GRU, which is our base model and, then we proceed to the details of our prior knowledge attention based BiGRU (ATBiGRU) model.

3.5.1 Sequential GRU

Gated recurrent unit (GRU) [8] is a popular RNN model, which has been extensively used in sentiment analysis to deal with long-term dependencies [18]. It sequentially takes each word in the input text sequence, models its information and produces a hidden state that contains contextual information. Firstly, let the feature maps $P = [p_1, p_2, p_3, \ldots, p_{k-1}, p_k] \in \mathbb{R}^{k \times \bar{d}}$ and $\bar{L}$

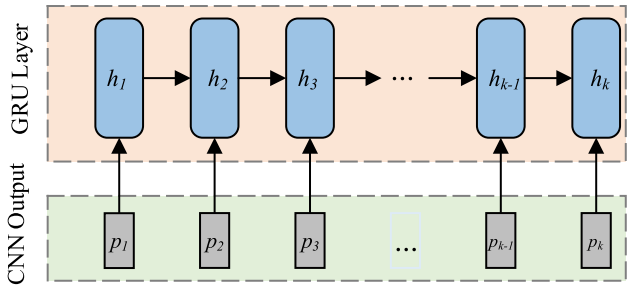

Fig. 2 Sequential GRU

be the input and the hidden state dimension of single direction GRU, respectively. As is shown in Fig. 2, at time step t, the hidden state $h_t \in \mathbb{R}^{\bar{L}}$ of a single direction GRU is computed as below:

$$r_t = \sigma(W_r p_t + U_r h_{t-1}) \tag{7}$$

$$z_t = \sigma(W_z p_t + U_z h_{t-1}) \tag{8}$$

$$\tilde{h}_t = \tanh(W_h p_t + U_h(r_t \odot h_{t-1})) \tag{9}$$

$$h_t = (1 - z_t)h_{t-1} + z_t \tilde{h}_t \tag{10}$$

where r_t, z_t, $\tilde{h}_t$, σ, $\odot$, and $\{U_* \in \mathbb{R}^{\bar{L}\times\bar{L}}, W * \in \mathbb{R}^{k\times\bar{L}}\}$ denote the reset gate, update gate, candidate activation, sigmoid function, Hadamard product, and weight matrices of a single directional GRU, respectively.

Then, to allow each word for containing the contextual information of its predecessor and successor words in the input text sequence, BiGRU is used. First, to get forward hidden state $\overrightarrow{h}_t$, it processes the input text sequence with the GRU in the forward direction with Eqs. (7)–(10). Similarity, by modeling the text sequence with the GRU in a backward direction, it updates the backward hidden state $\overleftarrow{h}_t$. Finally, the two hidden states $\overrightarrow{h}_t$ and $\overleftarrow{h}_t$ are combined as follows:

$$y_t = [\overrightarrow{h}_t \oplus \overleftarrow{h}_t] \tag{11}$$

where $t = 1, \ldots, k$, $\oplus$ denotes the element-wise sum between the forward and backward hidden state vectors. The hidden state vector $y_t \in \mathbb{R}^{\bar{L}}$ represents the global long-term dependency at time step t as it contains text sequence information from both directions. Therefore, depending on the required task to accomplish, one can either use the final hidden state $y_k \in \mathbb{R}^{\bar{L}}$ or whole output sequence of the BiGRU collected in a matrix $Y \in \mathbb{R}^{k\times\bar{L}}$ as follows:

$$Y = [y_1, y_2, y_3, \ldots, y_{k-1}, y_k] \tag{12}$$

However, the BiGRU's capacity to deal with global long-range dependencies can be limited by the long distance between dependencies as it processes the input in a sequential manner where the current time step can only access its successor or predecessor. Furthermore, it considers all words equally.

3.5.2 Prior knowledge attention based BiGRU

Inspired by recurrent skip connection [7, 47], context-aware LSTM [25], and conscience prior network [4], we address the above issues using prior knowledge ATBiGRU, whose single direction is illustrated in Fig. 3. The proposed ATBiGRU is a prior knowledge model in the sense that the current time step has access to all previous hidden states that serve as its context (prior knowledge).

To achieve the purpose, we introduce a global context memory (*GCM*), which stores the hidden states at each time step and supplies them to the current time step to bring

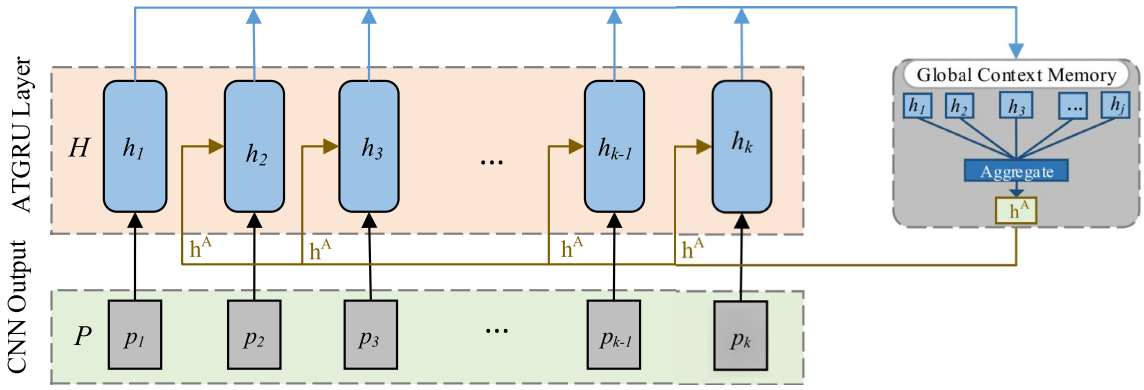

Fig. 3 Illustration of the proposed prior knowledge attention based GRU network. The golden lines indicate the aggregate (h^A) of previous hidden states. The blue lines mark the hidden states being loaded to the Global Context Memory (*GCM*)

closer the contextual information of the input text sequence. However, all hidden states don't equally contribute to the present time step, and all of them can not be supplied to the current time step due to the dimensions mismatch. Therefore, we apply the attention mechanism on *GCM* to produce the aggregated hidden state (h^A) that can be inputted to the current time step. Thus, at time step t, in contrast to the sequential GRU, which only accesses one prior hidden state h_{t-1}, our prior ATGRU updates the hidden state h_t with the aggregated hidden state (h^A) as below:

$$r_t = \sigma(W_r p_t + U_r h^A) \tag{13}$$

$$z_t = \sigma(W_z p_t + U_z h^A) \tag{14}$$

$$\tilde{h}_t = \tanh(W_h p_t + U_h(r_t \odot h^A)) \tag{15}$$

$$h_t = (1 - z_t)h^A + z_t \tilde{h}_t \tag{16}$$

where h^A is the aggregate of $j = t - 1$ previous hidden states, which is expressed by Eq. (19), and the most relevant hidden states to the present time step are assigned higher weights.

At time step t, when a new hidden state is loaded to *GCM*, the *GCM* produces the h^A based on the attention operations expressed as follows:

$$g_t = tanh(W_m h_t + b_m) \tag{17}$$

$$\alpha_t = \frac{\exp(g_t c_v)}{\sum_{t=1}^{j} \exp(g_t c_v)} \tag{18}$$

$$h^A = \sum_{t=1}^{j} \alpha_t h_t \tag{19}$$

where $h^A \in \mathbb{R}^{\bar{L}}$ is a vector that contains the weighted sum of $j = t - 1$ prior hidden states. g_t, α_t, c_v, W_m, and b_m denotes the hidden representation of h_t, normalized weight of the hidden state h_t, context vector, weight, and bias, repressively.

In the end, by processing the input text sequence in the forward and backward directions with our prior knowledge ATBiGRU, we get the final output $Z \in \mathbb{R}^{k \times \bar{L}}$ represented as follows:

$$Z = [z_1, z_2, z_3, \ldots, z_{k-1}, z_k] \tag{20}$$

where $z_t = [\overrightarrow{h_t} \oplus \overleftarrow{h_t}]$. Therefore, aiming to classify the input text sequence, we use the final hidden state $z_k \in \mathbb{R}^{\bar{L}}$ as is shown in Fig. 1. Acting in this way, z_k is considered to contain the global contextual information of the whole sequence since it has a short-cut connection to all previous hidden states.

Table 1 Summary statistics of the datasets

Dataset	#Train	#Val	#Test	#Classes
SSTb	8544	1101	2210	2
IMDB	25,000	–	25,000	2
Amazon1 P	166,314	81,916	81,916	2
Amazon1 F	186,697	91,956	91,956	5
Amazon2 P	667,155	328,599	328,599	2
Amazon2 F	735,376	362,201	362,201	5

3.6 Output layer

The output layer takes the hidden state vector $z_k \in \mathbb{R}^{\bar{L}}$ as input. Afterwards, the softmax is applied to estimate the probability distribution for each sentiment class label. Formally, the softmax operation is defined by:

$$P(y_i = c|b_i;w_c) = \frac{\exp(w_c^T z_k + b_i)}{\sum_{j=1}^{C} \exp(w_j^T z_k + b_j)} \tag{21}$$

where C, $b_i \in \mathbb{R}^C$, and $w_c \in \mathbb{R}^{\bar{L} \times C}$ denote number of classes, bias for class c, and weight for class c, respectively. We apply the cross-entropy loss to minimize the difference between the actual probability distribution and predicted the probability for each training sample:

$$L = -\sum_{i=1}^{C} t_c(y_i) logP(y_i = c|b_i;w_c) \tag{22}$$

where $t_c(y_i)$ is one-hot vector representing the distribution of the actual sentiment label and $P(y_i = c|b_i;w_c)$ is the predicted probability.

4 Experiments

We evaluate the effectiveness of ACGRN on several real-world datasets. Therefore, this section presents the empirical results obtained.

4.1 Dataset description

We evaluate the performance of the proposed model on the following six real-world sentiment analysis datasets. Their statistics are presented in Table 1. First, we adopt small datasets like IMDB Large Movie Review[1] [30] and Stanford Sentiment Treebank (SSTb)[2] [41]. We also consider

[1] http://ai.stanford.edu/~amaas/data/sentiment/.

[2] https://nlp.stanford.edu/sentiment/.

large datasets from Amazon review[3] [31]: Clothing, Shoes, and Jewelry Review dataset (we name this Amazon1), and CDs and Vinyl Review dataset (we name this Amazon2). For Amazon review datasets, we create two types of datasets. The first dataset has five labels (we name this F: Full). The second dataset is the sentiment polarity datasets (we name this P: Polarity), in which labels 1 and 2 are viewed as negative. Similarly, labels 4 and 5 are taken as positive. All these datasets present different properties. The number of examples varies from 8544 to 1 million. The number of classes is comprised of two to five classes. Besides, we bring to the readers' attention that for speeding up the training process, we fix the sentence length to 500 and 1024 on IMDB and Amazon datasets, respectively.

4.2 Experimental settings

The inputs to the proposed model are the embeddings initialized by GloVe[4] [37] with dimension 200 and they are updated with other parameters during the network training process. For the CNN model, we apply three channels where each one uses a one-dimensional convolutional layer with 256 filters(L), and the kernel window sizes $k(4, 5, 6)$. We use the rectifier linear unit (ReLU) activation function to each convolutional layer. Besides, each channel applies a max-pooling layer with size two. The hidden units ($\bar{L}$) for the GRU layer are fixed to 300. The number of epochs for training the proposed model on all datasets varies between (5, 7). For each iteration of the training process, we fix the batch size to 32. To prevent the model for overfitting, we apply the early stopping and dropout [42] with the dropout probability between 0.5 and 0.8 after the convolution layer. Adam optimizer [21] with default parameters is used to perform parameter optimization. While training the proposed model, we minimize the cross-entropy loss given by Eq. (22).

4.3 Baseline methods

We compare the effectiveness of the proposed model with CNNs, RNN with attention-based, and hybrids state-of-the-art deep learning approaches for sentiment classification. Furthermore, we compare our model with fastText [15], which is a simple but efficient baseline for text classification.

(1) *CNN-based approaches*

- CNN-Multi [20]: A model that uses two CNN channels with different filters to learn local contextual features.
- VDCNN [10]: A state-of-the-art very deep CNN proposed in text classification to deal with long-range dependencies.
- DeepCNN [12]: A deep CNN model that uses character information in the sentence representations.
- CNN-SA [56]: A simple CNN but with a better choice of hyper-parameters. It has been suggested in analyzing the sensitivity of CNN's components.

(2) *RNN with attention-based approaches*

- BiGRUATT [45]: A bidirectional gated recurrent unit coupled with attention mechanism suggested for capturing long-term dependencies.
- Tree-GRU [38]: A model that represents sentence information in the form of the tree. In this tree, the nodes are chosen based on the weight of each word computed using the attention mechanism.
- CBA+LSTM [27]: A Cognitive Based Attention LSTM approach that uses the attention mechanism built using the cognition ground eye-tracking data.
- HAN [53]: A hierarchical network with attention model, which has shown the strong performance of various review text datasets.

(3) *Hybrid approaches*

- CNN-LSTM [11]: A hybrid model that replaces the pooling layer of the CNN with an LSTM layer.
- DSCNN [55]: A dependency sensitivity CNN that hierarchically applies the LSTM to represent sentence then uses CNN in the features extraction.
- HRL [47]: A hybrid residual LSTM model that integrates the ResNet connection with LSTM to perform the sequence classification.
- RCNN [22]: An approach that learns the contextual features produced by RNN with a max-pooling layer.
- SA-SNN [57]: A self-attention sandwich neural network model, which integrates LSTM and CNN in a sandwich form. It employs a self-attention mechanism to fully use the extracted local semantic and global structure representations.
- GLMA [29]: A global-local mutual attention approach, which extracts local contextual features with CNN and global contextual features with LSTM, then integrates them through a mutual attention mechanism.

4.4 Model comparison with baseline methods

The experimental results achieved by ACGRN against baseline models on six real-world datasets are given in Table 2. Foremost, the experimental results reveal that ACGRN significantly outperforms all baseline methods for five out of six datasets.

(1) *Comparison with CNN-based approaches* In comparison against CNN-based models, the empirical results show

[3] http://jmcauley.ucsd.edu/data/amazon/.

[4] http://nlp.stanford.edu/projects/glove/.

Table 2 Experimental results [in accuracy] on SSTb, IMDB, Amazon1 P, Amazon1 F, Amazon2 P and Amazon2 F

Category	Model	SSTb (%)	IMDB (%)	Amazon1 P (%)	Amazon1 F (%)	Amazon2 P (%)	Amazon2 F (%)
CNN-based	CNN-multi	88.90	87.26	93.60	86.30	91.15	87.74
	CNN-SA	88.90	88.10	92.80	87.10	92.30	87.40
	VDCNN	89.30	88.70	94.80	87.08	93.79	87.87
	DeepCNN	85.22	89.11	92.60	87.17	94.30	87.28
RNN-att-based	BiGRUATT	89.30	89.91	94.80	88.79	95.30	89.20
	CBA+LSTM	89.50	90.70	94.60	88.50	94.80	88.74
	HAN	89.34	89.96	94.71	88.58	94.51	87.56
	Tree-GRU	89.50*	–	–	–	–	–
Hybrids	RCNN	89.70	90.10	93.97	88.69	95.10	87.54
	DSCNN	89.43	90.66	95.20	88.87	96.10	88.95
	CNN-LSTM	89.10	90.75	89.33	83.41	90.71	83.30
	SA-SNN	89.16	91.13	95.17	88.34	95.21	88.87
	GLMA	89.72	91.51	95.87	89.27	96.22	**89.43**
	HRL	–	90.92*	–	–	–	–
Linear model	fastText	88.78	90.03	94.28	88.14	94.02	87.56
Ours	ACGRN	**89.90**	**91.74**	**96.12**	**89.87**	**96.40**	89.20

For our experiments, we report the mean accuracy of 5 runs. The results with * are reported from their original paper. The best performances are in bold

that ACGRN consistently outperforms them with an absolute improvement in accuracy by 0.6%, 2.63%, 1.32%, 2.7%, 2.1%, and 1.33% on SSTb, IMDB, Amazon1 P, Amazon1 F, Amazon2 P, and Amazon2 F, respectively. It is evidence that ACGRN is more efficient in modeling contextual features compared to individual CNN-based models. Besides, this increase in accuracy justifies the potential of using prior knowledge attention based BiGRU that helps in modeling global contextual features.

(2) *Comparison with RNN with attention-based approaches* In comparison with RNN with attention-based models, we observe that ACGRN performs better than them with an increase in accuracy of 0.4%, 1.04%, 1.32%, 1.08%, and 1.1% on SSTb, IMDB, Amazon1 P, Amazon1 F, and Amazon2 P, respectively. In this setting, the outstanding performance of the ACGRN model is first attributed to local contextual features produced by the proposed CNN. Second, we attribute it to the global contextual features learned by the proposed prior knowledge ATBiGRU. Therefore, it shows the benefits of applying the attention mechanism to each time step instead of waiting for the last output of GRU. Moreover, the results reveal that the cognition-based attention in CBA+LSTM does not make a significant difference with our proposed attention, which is based on the context of a word in the text sequence.

(3) *Comparison with hybrid approaches* By comparing ACGRN with hybrid approaches, we observe that ACGRN outperforms these models with an improvement in accuracy by 0.18%, 0.23%, 0.25%, 0.6%, and 0.18% on SSTb, IMDB, Amazon1 P, Amazon1 F, Amazon2 P, respectively. We attribute the better performance of ACGRN to the best choice of combining CNN with prior knowledge attention based BiGRU that allows the current time step to have access to all previous hidden states. In particular, the attention mechanism allows our model to focus on the salient words that hold sentiment information. Also, we realize that baseline CNN-LSTM performs worse on large datasets because it does not use the pooling layer.

In comparison against the fastText model, we observe that ACGRN outperforms it by 1.2%, 1.71%, 1.84%, 1.73%, 2.38%, and 1.64% in terms of accuracy on SSTb, IMDB, Amazon1 P, Amazon1 F, Amazon2 P, and Amazon2 F, respectively. Therefore, these results demonstrate that ACGRN is more efficient in modeling contextual features compared with fastText. However, fastText presents the advantage of being fast compared to our models as well as other baseline methods.

In brief, the experimental results prove our initial hypothesis that using local and global contextual features improves the performance accuracy for sentiment classification.

4.5 Ablation studies

We perform ablation studies to evaluate the technical contributions of the ACGRN's components. Thus, a set of ACGRN's variants and the empirical results achieved are presented in Table 3. Note that, the models in this subsection follows the experiment setup in Sect. 4.2.

The effect of contextual features' complement To evaluate the significance of using both types of contextual features, i.e., local contextual features and global contextual

Table 3 Ablation studies of different components of ACGRN

Variant	SSTb (%)	IMDB (%)	Amaz1 P (%)	Amaz1 F (%)
MCNN	89.1	90.58	93.70	83.41
sGRU	88.6	91.06	93.40	83.32
BiGRU	89.1	90.88	93.10	86.60
CsGRU	89.2	90.74	89.33	83.41
CBiGRU	89.7	91.39	94.12	88.22
ATGRU	88.9	90.60	94.70	88.27
ATBiGRU	89.5	91.33	95.10	89.17
ATCsGRU	89.5	91.09	94.77	88.47
CGRN+ATT	89.5	91.21	95.34	88.97
ACGRN	**89.9**	**91.74**	**96.12**	**89.87**

Amaz means Amazon. The results are in accuracy. The best performances are in bold

features, we compare our hybrid models, i.e., CsGRU, CBiGRU, ATCsGRU, and ACGRN against their component models. Generally, the combined approaches significantly outperform their component models across all datasets, which shows the advantage of exploiting both type contextual features. For instance, when we consider the ACGRN's variant models, i.e., multi-channel CNN (MCNN) that uses only local contextual features and prior knowledge ATBiGRU that makes use of global contextual features to perform sentiment classification, the results indicate that ACGRN outperforms ATBiGRU by 0.4%, 0.41%, 1.02%, and 0.7% on SSTb, IMDB, Amazon1 P and Amazon1 F, respectively. Similarly, it outweighs MCNN by 0.8%, 1.16%, 2.42%, and 6.46% on SSTb, IMDB, Amazon1 P, and Amazon1 F, respectively.

The effect of global context memory To assess the advantage offered by GCM, we first compare our models that contain the GCM components against their counterparts, which do not include it. We observe that our models with GCM, i.e., ATGRU, ATBiGRU, ATCsGRU, and ACGRN, significantly perform well compared to their counterparts without GCM, i.e., GRU, BiGRU, CsGRU, and CBiGRU, respectively. For instance, our top-performing model, i.e., ACGRN, outperforms CBiGRU with an improvement of 0.2%, 0.35%, 2%, and 1.65% in terms of accuracy on SSTb, IMDB, Amazon1 P, and Amazon1 F, respectively. Thus, the results evidence their better skills to model global contextual features using the GCM. Besides, we compare ACGRN against CGRN+ATT. Unlike ACGRN that uses GCM to compute the attention at each time step to bring closer the contextual information, in CGRN+ATT, the attention mechanism is applied to the whole output of BiGRU. The results show that ACGRN exceeds CGRN+ATT with an improvement in accuracy by 0.4%, 0.53%, 0.78%, and 0.9% on SSTb, IMDB, Amazon1 P, and Amazon1 F, respectively. It confirms the usefulness of attention mechanism in prioritizing informative features at each time step rather than waiting for the last output of the GRU.

Table 4 Comparison results of aggregation methods

Variant	SSTb (%)	IMDB (%)	Amaz1 P (%)	Amaz1 F (%)
MaxCGRN	88.31	91.44	95.25	88.94
AvgCGRN	88.43	91.49	95.34	89.24
AddCGRN	88.89	91.18	94.87	88.78
ACGRN	**89.9**	**91.74**	**96.12**	**89.87**

Amaz means Amazon. The results are in accuracy. The best performances are in bold

The effect attention mechanism as an aggregation method To verify the effectiveness of the attention mechanism in aggregating the previous hidden states in our model, we design MaxCGRN, AvgCGRN, AddCGRN models by replacing the attention mechanism with max-pooling, average pooling, and addition aggregation methods, respectively. In Table 4, we present the comparison results among ACGRN, MaxCGRN, AvgCGRN, and AddCGRN. It shows that our ACGRN model with attention mechanism outperforms its counterparts across all datasets. Thus, the results demonstrate the effectiveness of the proposed attention mechanism in aggregating the most discriminative and informative hidden states into a single vector representation.

The effect of GRU's direction To evaluate the advantages of modeling the input text sequence in the forward and backward directions, we compare our model variants that contain a bidirectional layer against their counterparts without it. Looking at the results, we observe that our models with bidirectional layer, i.e., BiGRU, ATBiGRU, CBiGRU, and ACGRN consistently outperform their counterparts without it, i.e., sGRU, ATGRU, CsGRU, and ATCsGRN, respectively. Therefore, the results confirm the advantages of modeling the input in both directions as a way of enriching the global contextual features extracted by GRU.

In a nutshell, based on the analysis of the results of ablation studies on ACGRN's variants, we stress that the changes brought by global context memory give good results when coupled with BiGRU. This assumption remains true across all datasets considered in the experiments. Also, to get good results, the features learned by CNN need to be modeled with ATBiGRU. Besides, it is worth mentioning that the majority of ACGRN's variants outperforms some of the baselines.

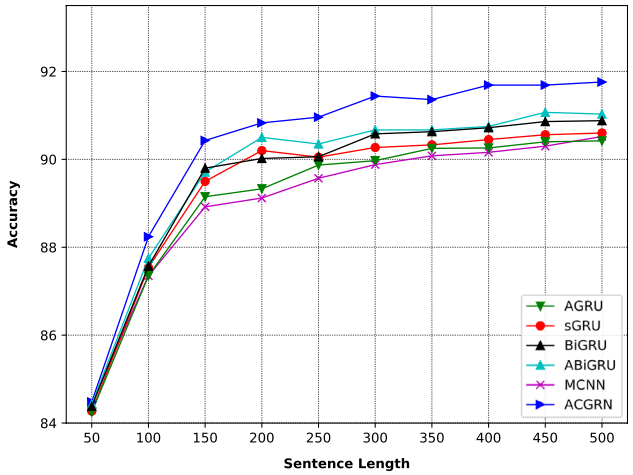

Fig. 4 Sentiment classification accuracy versus sentence length on IMDB dataset

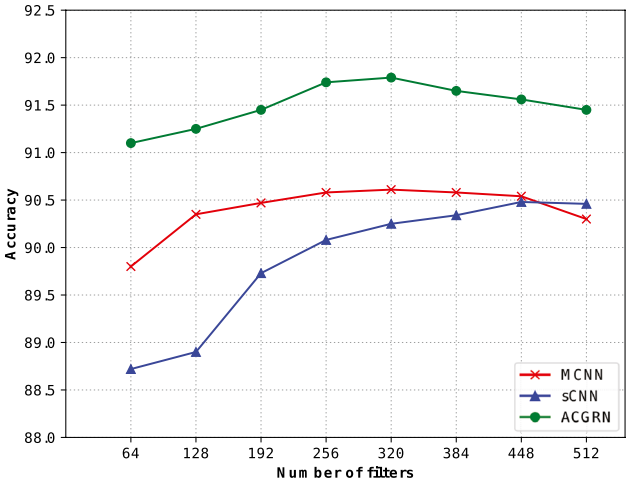

Fig. 6 Sentiment classification accuracy versus number of filters on IMDB dataset

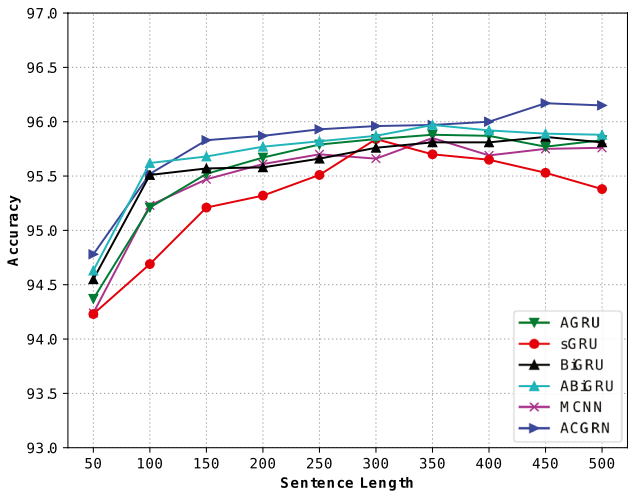

Fig. 5 Sentiment classification accuracy versus sentence length on Amazon1 P dataset

5 Discussion and qualitative analysis

5.1 Effect of sentence length

One hypothesis to explain the advantages of ACGRN over CNN and GRU is that it helps to learn contextual features of short and long sentences. If this were true, ACGRN could consistently outperform both CNN and GRU on short and long sentences. Therefore, to facilitate the interpretation, we illustrate the relationship between sentence length and the performance accuracy in Figs. 4 and 5. Figures 4 and 5 show the performance achieved by MCNN, sGRU, BiGRU, ATGRU, ATBiGRU, and ACGRN on IMDB and Amazon1 P datasets, respectively. We bring to the readers' attention that the models in this subsection follow the experimental protocol described in Sect. 4.2. Besides, we vary the sentence length of both datasets considered in this section from 50 up to 500.

Foremost, we observe that ACGRN consistently outperforms its counterparts across all sentence lengths on both datasets. Also, from Fig. 4, we observe that the rate of performance increase of ACGRN and ATBiGRU models on IMDB is high compared to other variants from sentence length equal to three hundred. On the other side, from Fig. 5, we observe the capability of our model variants for modeling long sentences. The direct observation is that the performance accuracy of ACGRN keeps increasing up to sentence length equal to 450. Besides, with support of ablation study experimental results in Table 3 with the sentence length fixed to 1024, we observe that the performance accuracy of all models decreases. However, the rate of decrease of ACGRN's performance accuracy is low compared to its variants. For example, the accuracy achieved by ACGRN is 96.14% and 96.12% when the sentence length is equal to 500 and 1024, respectively. Whereas, the performance accuracy achieved by MCNN drops from 95.76% to 93.7% for the sentence length is equal to 500 and 1024, respectively. Therefore, these results show that ACGRN performs well on short and long sentences.

5.2 Effect of filter numbers

The increase of filter number does not always result in accuracy increase rather than can increase model parameters. In this subsection, we experiment by varying the numbers of filters in our models that contain the convolution layer. Thus, we present the performance accuracy achieved on IMDB and Amazon1 P datasets in Figs. 6 and 7, respectively. Note that,

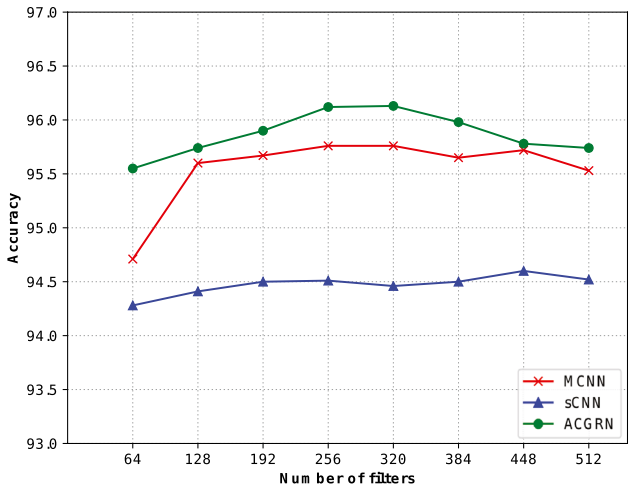

Fig. 7 Sentiment classification accuracy versus number of filters on Amazon1 P dataset

the experiments in this subsection follow the experimental protocol described in Sect. 4.2.

From the experimental results, we observe that the increasing number of filters does not always achieve the best accuracy. The results indicate that performance achieved by ACGRN on both datasets peaks when the number of filters is between 256 and 320. On the other hand, ACGRN overfits if the number of filters increases to more than 320. This overfitting is caused by the increase of filter numbers that can result in a large number of model parameters, which lead to model overfitting. In brief, this behavior of ACGRN of using fewer filters to achieve excellent performance is attributed to our multi-channel CNN, which is shallow and wide.

5.3 Case study for visualization of attention

To illustrate the advantages of our ACGRN model over CNN and RNN, we visualize how the attention focuses on the most contributing words. The attention visualization for several review texts from the Amazon1 dataset is demonstrated in Fig. 8. Note that the darker color means higher weight.

Considering the following positive review text shown in Fig. 8a. ACGRN does not only take into account words carrying strong sentiment like "disappointing", "good", "very" but also deals with the context across the sentence. In this example, by looking at the first part of the sentence before the word "but", one may think that the sentence is negative since it contains sub-sentence "it doesn't have" with "negation" n-gram "doesn't". However, our model looks at the whole context of the review text and finds that the second part of the sentence is very positive since it contains words like "very", "good" and their corresponding n-grams. Hence, our model correctly predicts the review as positive.

Taking into a negative review text presented in Fig. 8b1, b2, this review text has a long and complicated structure that it is difficult for simple CNN or RNN to obtain the correct label. Therefore, ACGRN obtains the correct negative label with the help of sentence global contextual features.

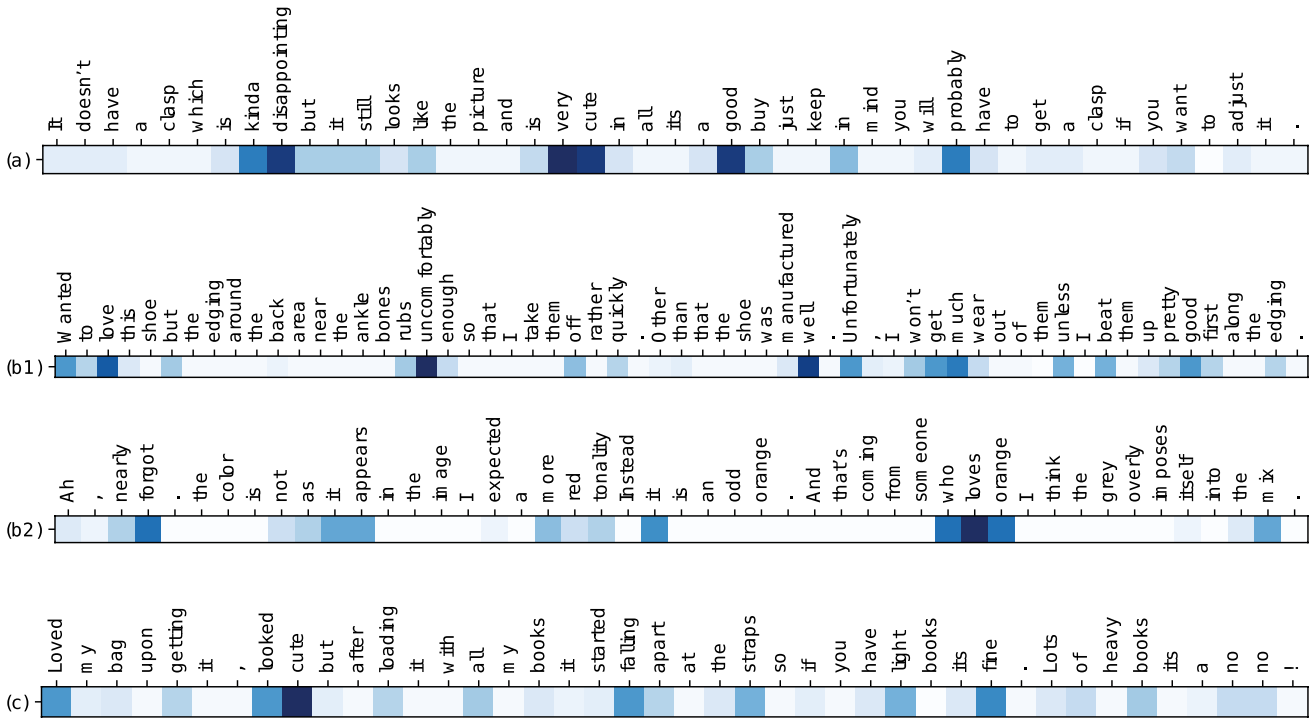

Fig. 8 Visualization of attention over the words from review sentences in Amazon1 dataset

However, there is no one size fits all. ACGRN fails to obtain the correct label in some instances. Let us consider the following negative review text shown in Fig. 8c. ACGRN obtains the positive label because it requires modeling different aspects of the object to find real sentiment. In this case, the sentiment of the review text with respect to the bag is positive, while the problem arises when it comes to judging its weight.

6 Conclusion

In this paper, we propose a novel model ACGRN to learn local and global contextual features. Specifically, we apply three concurrent convolutional layers followed by max-pooling to extract a possible number of local contextual features. Subsequently, we use our novel prior knowledge ATBiGRU to learn global contextual features from the representations encoded by the CNN layer. The proposed prior knowledge ATBiGRU accesses all previous hidden states as an aggregated hidden state generated by the attention mechanism. We have evaluated the effectiveness of the ACGRN model on six small and large real-world datasets. The proposed ACGRN consistently outperforms the state-of-the-art methods.

The experimental results indicate that ACGRN improved the performance accuracy by 0.18%, 0.23%, 0.25%, 0.6%, and 0.18% on SSTb, IMDB, Amazon1 P, Amazon1 F, and Amazon2 P, respectively. Furthermore, the ablation studies' experimental results allow us to stress that the BiGRU modified by the attention mechanism gives good results compared to single-channel GRU modified by the attention mechanism. The qualitative analysis shows that ACGRN is good at learning short and long sentences. Thus, this work validates our idea of using contextual features for better performance in sentiment analysis.

This research can be extended in multiple dimensions. Therefore, direct future work is to incorporate our model into an end-to-end manner to other natural language tasks to solve the problem of long-range dependencies. In particular, we would like to investigate the effectiveness of our model on multilingual sentiment analysis with text sequences that may contain long-term dependencies.

Acknowledgements This work is supported by the National Key Research and Development Program of China under Grants 2016YFB0800402 and 2016QY01W0202, National Natural Science Foundation of China under Grants U1836204, U1936108, 61433006, U1401258, and 61502185.

References

1. AlSmadi M, Talafha B, AlAyyoub M, Jararweh Y (2019) Using long shortterm memory deep neural networks for aspect based sentiment analysis of Arabic reviews. Int J Mach Learn Cybern 10(8):2163–2175
2. Amplayo RK, Kim J, Sung S, Hwang S (2018) Cold-start aware user and product attention for sentiment classification. In: Proceedings of the 56th annual meeting of the association for computational linguistics (ACL), pp 2535–2544
3. Bahdanau D, Cho K, Bengio Y (2015) Neural machine translation by jointly learning to align and translate. In: 3rd international conference on learning representations (ICLR), pp 1–15
4. Bengio Y (2017) The consciousness prior. CoRR arXiv:1709.08568
5. Cai Y, Yang K, Huang D, ZhouXue Z, Lei X, Xie H et al (2019) A hybrid model for opinion mining based on domain sentiment dictionary. Int J Mach Learn Cybern 10(8):2131–2142
6. Cambria E, White B, Durrani TS, Howard N (2014) Computational intelligence for natural language processing [guest editorial]. IEEE Comput Intell Mag Nat Lang Process 9(1):19–63
7. Campos V, Jou B, Giró i Nieto X, Torres J, Chang S (2018) Skip RNN: learning to skip state updates in recurrent neural networks. In: 6th international conference on learning representations (ICLR), pp 1–17
8. Cho K, van Merriënboer B, Gülçehre C, Bahdanau D, Bougares F, Schwenk H et al (2014) Learning phrase representations using RNN encoder–decoder for statistical machine translation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 1724–1734
9. Collobert R, Weston J (2008) A unified architecture for natural language processing: deep neural networks with multitask learning. In: Proceedings of the twenty-fifth international conference machine learning (ICML), pp 160–167
10. Conneau A, Barrault L, Schwenk H, LeCun Y (2017) Very deep convolutional networks for text classification. In: Proceedings of the 15th conference of the European chapter of the association for computational linguistics (EACL), pp 1107–1116
11. Hassan A, Mahmood A (2017) Deep learning approach for sentiment analysis of short texts. In: 3rd international conference on control, automation and robotics (ICCAR), pp 705–710
12. dos Santos CN, Gatti M (2014) Deep convolutional neural networks for sentiment analysis of short texts. In: 25th international conference on computational linguistics (COLING), pp 69–78
13. Hemmatian F, Sohrab MK (2019) A survey on classification techniques for opinion mining and sentiment analysis. Artif Intell Rev 52(3):1495–1545
14. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
15. Joulin A, Grave E, Bojanowski P, Mikolov T (2017) Bag of tricks for efficient text classification. In: Proceedings of the 15th conference of the European chapter of the association for computational linguistics, pp 427–431
16. Johnson R, Zhang T (2015) Effective use of word order for text categorization with convolutional neural networks. In: The 2015 conference of the North American chapter of the association for computational linguistics: human language technologies (HLT-NAACL), pp 103–112
17. Johnson R, Zhang T (2017) Deep pyramid convolutional neural networks for text categorization. In: Proceedings of the 55th annual meeting of the association for computational linguistics (ACL), pp 562–570

18. Habimana O, Li Y, Li R, Gu X (2020) Sentiment analysis using deep learning approaches: an overview. Sci China Inf Sci 63(1):111102
19. Kalchbrenner N, Grefenstette E, Blunsom P (2014) A convolutional neural network for modelling sentences. In: Proceedings of the 52nd annual meeting of the association for computational linguistics (ACL), pp 655–665
20. Kim Y (2014) Convolutional neural networks for sentence classification. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 1746–1751
21. Kingma DP, Ba J (2015) Adam: a method for stochastic optimization. In: 3rd international conference on learning representations (ICLR), pp 1–15
22. Lai S, Xu L, Liu K, Zhao J (2015) Recurrent convolutional neural networks for text classification. In: Proceedings of the twenty-ninth AAAI conference on artificial intelligence, pp 2267–2273
23. Le HT, Cerisara C, Alexandre DA (2018) Do convolutional networks need to be deep for text classification? In: The workshops of the thirty-second AAAI conference on artificial intelligence, pp 29–36
24. Liu B (2012) Sentiment analysis and opinion mining. Synthesis lectures on human language technologies. Morgan & Claypool Publishers, San Rafael
25. Liu J, Wang G, Hu P, Duan LY, Kot AC (2017) Global context-aware attention LSTM networks for 3D action recognition. In: IEEE conference on computer vision and pattern recognition (CVPR), pp 3671–3680
26. Long F, Zhou K, Ou W (2019) Sentiment analysis of text based on bidirectional LSTM with multi-head attention. IEEE Access 7:141960–141969
27. Long Y, Qin L, Xiang R, Li M, Huang C (2017) A cognition based attention model for sentiment analysis. In: Proceedings of the 2017 conference on empirical methods in natural language processing (EMNLP), pp 462–471
28. Luong T, Pham H, Manning CD (2015) Effective approaches to attention-based neural machine translation. In: Proceedings of the 2015 conference on empirical methods in natural language processing (EMNLP), pp 1412–1421
29. Ma Q, Yu L, Tian S, Chen E, Ng WWY (2019) Global-local mutual attention model for text classification. IEEE/ACM Trans Audio Speech Lang Process 27(12):2127–2139
30. Maas AL, Daly RE, Pham PT, Huang D, Ng AY, Potts C (2011) Learning word vectors for sentiment analysis. In: Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies (ACL), pp 142–150
31. McAuley JJ, Leskovec J (2013) Hidden factors and hidden topics: understanding rating dimensions with review text. In: Seventh ACM conference on recommender systems (RecSys), pp 165–172
32. Mishra A, Tamilselvam S, Dasgupta R, Nagar S, Dey K (2018) Cognition-cognizant sentiment analysis with multitask subjectivity summarization based on annotators' gaze behavior. In: Proceedings of the 32nd AAAI conference on artificial intelligence, pp 5884–5891
33. Mousa AE, Schuller BW (2017) Contextual bidirectional long short-term memory recurrent neural network language models: A generative approach to sentiment analysis. In: Proceedings of the 15th conference of the European chapter of the association for computational linguistics (EACL), pp 1023–1032
34. Muhammad A, Wiratunga N, Lothian R (2016) Contextual sentiment analysis for social media genres. Knowl Based Syst 108:92–101
35. Mujika A, Meier F, Steger A (2017) Fast-slow recurrent neural networks. In: Advances in neural information processing systems 30: annual conference on neural information processing systems (NIPS), pp 5917–5926
36. Pang B, Lee L (2007) Opinion mining and sentiment analysis. Found Trends Inf Retr 2(1–2):1–135
37. Pennington J, Socher R, Manning CD (2014) Glove: global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 1532–1543
38. Potamianos A, Kokkinos F (2017) Structural attention neural networks for improved sentiment analysis. In: Proceedings of the 15th conference of the European chapter of the association for computational linguistics (EACL), pp 586–591
39. Pozzi FA, Fersini E, Messina E, Liu B (2016) Sentiment Analysis in Social Networks. Morgan Kaufmann Publishers Inc
40. Qiao X, Peng C, Liu Z, Hu Y (2019) Word-character attention model for Chinese text classification. Int J Mach Learn Cybern 10(12):3521–3537
41. Socher R, Perelygin A, Wu J, Chuang J, Manning CD, Ng AY, Potts C (2013) Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of the 2013 conference on empirical methods in natural language processing (EMNLP), pp 1631–1642
42. Srivastava N, Hinton GE, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15(1):1929–1958
43. Tai KS, Socher R, Manning CD (2015) Improved semantic representations from tree-structured long short-term memory networks. In: Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing of the Asian federation of natural language processing (ACL–AFNLP), pp 1556–1566
44. Wang J, Yu L, Lai KR, Zhang X (2019) Investigating dynamic routing in tree-structured LSTM for sentiment analysis. In: Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing, pp 3430–3435
45. Wang N, Wang J, Zhang X (2017) YNU-HPCC at IJCNLP-2017 task 4: attention-based bi-directional GRU model for customer feedback analysis task of English. In: Proceedings of the IJCNLP, pp 174–179
46. Wang L, Tu Z, Way A, Liu Q (2017) Exploiting cross-sentence context for neural machine translation. In: Proceedings of the 2017 conference on empirical methods in natural language processing (EMNLP), pp 2826–2831
47. Wang Y, Tian F (2016) Recurrent residual learning for sequence classification. In: Proceedings of the 2016 conference on empirical methods in natural language processing (EMNLP), pp 938–943
48. Weston J, Chopra S, Bordes A (2015) Memory networks. In: 3rd international conference on learning representations (ICLR), pp 1–15
49. Wilson T, Wiebe J, Hoffmann P (2005) Recognizing contextual polarity in phrase-level sentiment analysis. In: Human language technology conference and conference on empirical methods in natural language processing, proceedings of the conference (HLT/EMNLP), pp 347–354
50. Wu Z, Dai X, Yin C, Huang S, Chen J (2018) Improving review representations with user attention and product attention for sentiment classification. In: Proceedings of the thirty-second AAAI conference on artificial intelligence, (AAAI-18), the 30th innovative applications of artificial intelligence (IAAI-18), and the 8th AAAI symposium on educational advances in artificial intelligence (EAAI-18), pp 5989–5996
51. Xu G, Meng Y, Qiu X, Yu Z, Wu X (2019) Sentiment analysis of comment texts based on BiLSTM. IEEE Access 7:51522–51532
52. Yang M, Tu W, Wang J, Xu F, Chen X (2017) Attention based LSTM for target dependent sentiment classification. In:

Proceedings of the thirty-first AAAI conference on artificial intelligence, pp 5013–5014
53. Yang Z, Yang D, Dyer C, He X, Smola AJ, Hovy EH (2016) Hierarchical attention networks for document classification. In: The 2016 conference of the North American chapter of the association for computational linguistics: human language technologies, pp 1480–1489
54. Zhang M, Zhang Y, Vo D (2016) Gated neural networks for targeted sentiment analysis. In: Proceedings of the thirtieth AAAI conference on artificial intelligence (AAAI), pp 3087–3093
55. Zhang R, Lee H, Radev DR (2016) Dependency sensitive convolutional neural networks for modeling sentences and documents. In: The 2016 conference of the North American chapter of the association for computational linguistics: human language technologies (NAACL/HLT), pp 1512–1521
56. Zhang Y, Wallace BC (2017) A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. In: Proceedings of the the 8th international joint conference on natural language processing (IJCNLP), pp 253–263
57. Zhao J, Zhan Z, Yang Q, Zhang Y, Hu C, Li Z et al (2018) Adaptive learning of local semantic and global structure representations for text classification. In: Proceedings of the 27th international conference on computational linguistics (COLING), pp 2033–2043
58. Zheng L, Wang H, Gao S (2018) Sentimental feature selection for sentiment analysis of Chinese online reviews. Int J Mach Learn Cybern 9(1):75–84

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.