CONVOLUTIONAL WITH ATTENTION GATED RECURRENT NETWORK FOR SENTIMENT ANALYSIS

By Olivier HABIMANA, Innocent KABANDANA & Alfred UWITONZE

School of Computer Science, Kigali Independent University, Kigali, Rwanda

ABSTRACT

In recent years, deep learning approaches like convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have received much attention to natural language processing tasks, especially to sentiment analysis. Different methods will be used to measure the Convolutional with Attention Gated Recurrent Network for Sentiment Analysis. Thankfully, these methods achieved significant results. However, these approaches individually fail to accomplish the task of sentiment analysis at the extent level. In sentiment analysis, the likelihood of a given word is estimated based on long-term dependencies and local contextual features that depend on a word and its neighboring words. This paper suggests a Convolutional with Attention Gated Recurrent Network (CAGRN) model performs the sentiment analysis by extracting these features. The objective behind our model is to apply the CNN layer to extract local contextual features. Afterward, the CAGRNusesa bidirectional gated recurrent unit (Bi-GRU) layer to encode the long-term dependence features. On the other hand, the attention mechanism is applied to help our model select the convenient words that hold sentiment information. The CAGRN performs better in sentiment

analysis by using the learned features. Our approach achieves competitive results on two real datasets IMDb and SSTb, compared with baseline models and requires fewer parameters. Executing various ablation experiments of our model components will be done in future.

Key findings: Convolutional Neural Networks (CNNs); Recurrent Neural Networks (RNNs); Convolutional with Attention Gated Recurrent Network (CAGRN)

1 Introduction

Web 2.0 applications, such as online social networking and e-commerce websites, have exploded in popularity recently, allowing participants to freely share their ideas and opinions in a text(Pang & Lee, 2005; Pozzi et al., 2017). Sentiment analysis is a natural language processing task that looks for opinions expressed in user-generated content (UGC). Discovering hidden knowledge from user-generated content (UGC) is priceless(Liu, 2012; Pang & Lee, 2008) to individual-level up to big organizations and governments. An individual user can decide to buy a product by judging other customers' comments who have purchased that product. By analyzing customers' reviews, e-commerce companies can improve

their service delivery. Government organizations can take different measures based on the understanding of the public opinions about any trending topic. Additionally, sentiment analysis can improve the capability of recommender systems by identifying the aspects that the user wants(Poria et al., 2016; Z. Wang & Zhang, 2017).

So far, numerous approaches for sentiment analysis have been proposed in the literature (Liu, 2012; Pozzi et al., 2017; L. Zhang et al., 2018). The designed approaches extract and apply important features in sentiment analysis. However, the suggested approaches perform the sentiment analysis by relying on the general features extracted from the input embeddings. This is certainly helpful, but it is not always a perfect solution in sentiment analysis. In practice, it is more important to perform the sentiment analysis by using all the contextual features of the word in a sentence, which we refer to as contextual sentiment analysis in this work.

Generally, in sentiment analysis, the likelihood of a given theword is estimated based on features that depend on a word and its neighboring words (Mousa & Schuller, 2017; Muhammad et al., 2016; Wilson et al., 2005). In this work, we focus on two important categories of features. The first category includes local contextual elements heavily influenced by the arrangement of words in a phrase. Actually, the order is important because the polarity of a

word in a sentence can change based on where it is in the sentence. Long-term dependencies are the second sort of feature that can exist in a sentence. Therefore, we claim that modeling these contextual features is of great value.

A natural method to solving the challenge of sentiment analysis is to use classic sentiment analysis approaches based on lexicons (Taboada et al., 2011), n-gram, and part-of-speech tags (POS) (Bespalov et al., 2011; Pang et al., 2002). Bag-of-words (BoW) (S. Wang & Manning, 2012) approaches can also be applied. However, the performance of these approaches in sentiment analysis is often unsatisfactory due to the following reasons. First, the performance of these approaches relies on tedious feature engineering work. Second, concerning lexicon-based approaches, the context sentiment of a givenword can be different from the prior polarity of that word in the lexicon (Muhammad et al., 2016). Third, the n-gram based models are accused of suffering from data sparseness. Lastly, BoW based approaches handle the input texts as unordered sets of words. Thus, they cannot model the necessary information and syntactic features for sentiment analysis

Furthermore, deep learning methods can be used to address the problem of contextual sentiment analysis. In recent years, thankfully, these approaches have improved the results considerably

in sentiment analysis due to the capability of automatic feature learning with a hierarchy of layers (Deng & Yu, 2013). Also, their success is attributed to the success of word embedding models that allow the distributed representation of words (Mikolov et al., 2013; Pennsington et al., 2014). Deep learning models like convolutional neural networks (CNNs) (Collobert & Weston, 2008) and recurrent neural networks (RNNs) like long short term memory (LSTM) (Hochreiter & Schmidhuber, 1997)and gated recurrent unit (GRU) (Cho et al., 2014)achieved tremendous success in sentiment analysis compared to other models. Consequently, numerous approaches have been proposed in the literature.

Researchers have suggested a plethora of CNNs based models for sentiment analysis; Kim (2014) used a multi-channel CNN for capturing multiple featuresinthe local context. Very deep CNNs, on the other hand, have been investigated for capturing long-range relationships(Conneau et al., 2017; Johnson & Zhang, 2017). Also, Johnson & Zhang (2014) applied a CNN-based model for the best use of word order to represent the text. Similarly, Kalchbrenner et al.(2014) investigated the use of dynamic CNN to learn the semantic features of a sentence. However, the proposed CNN-based models are a partial solution to the contextual sentiment analysis because CNN can only exploit the local features. In addition,

capturing long-range dependencies requires the CNN to be deeper; hence, expensive computational resources are required.

On the other hand, RNN based models proved to be efficient in learning the sequence inputs and modeling long-range dependencies to maintain the constant error flow (Mujika et al., 2017). As a result, to deal with the context in which the term appears, Lin et al. (2018) applied the structure-attention LSTM, Mousa & Schuller (2017) explored the use of a Bi-LSTM. Similarly, M. Zhang et al. (2016) suggested a gated RNN. Moreover, RNNs have been applied for capturing long-range dependencies, Yang et al. (2017) applied an LSTM. Similarly, Mujika et al. (2017) applied a Fast-Slow RNN. Likewise, N. Wang et al.(2017)suggested a Bi-GRU with attention, and also Chen et al.(2017) applied a multiple attention LSTM. Although these models produced interesting results, their behavior to sentiment analysis is still unsatisfactory. Their unsatisfactory performance is associated with the RNNs model the sentence in temporal order, i.e., the output depends on the previous context. In addition, RNNs do not preserve the structure of the input sequence. Furthermore, RNNs are biased in terms of the representation where the words at the beginning of the sentence are less considered than those at the end of the sentence. Thus, the RNN does not model the semantic information as it can appear anywhere in the sentence.

Therefore, a question raises to our mind: "How can we design a computationally less expensive model suitable for contextual sentiment analysis. A model that uses the contextual information at an extent level in that the order of the inputs is preserved; the local and semantic features are exploited; and the global features are captured".

To this end, motivated by the above findings, we suggested an approach called Convolutional Attention Gated Recurrent Network (CAGRN) to answer the above question. The CAGRN combines the CNN with Bidirectional-GRU (Bi-GRU) with an attention mechanism based. The CAGRN inherits the characteristics of CNN for preserving the spatial structure of the input sequence by using the one-dimensional structure of the text data (Johnson and Zhang, 2014), good local feature detectors, i.e., filters that capture n-gram at every position of the sentence and using few parameters that help speed up the training process. To avoid the deeper network, our model follows the multi-channel CNN model (Kim, 2014), which is shallow and wide. To model the input sentence in sequential order and capture the global features like long-term dependencies, we use the Bi-GRU that processes the input sentence forward and backward. Finally, we utilize the attention mechanism extensively applied in neural translation machines (Bahdanau et al., 2015;

Luong et al., 2015) to allow our model to prioritize the words containing the sentiment at any location in the sentence.

Overall, the main contributions of this paper are three-fold:

- We propose a combined approach CAGRNthat enhances the performance of CNN with Bi-GRU coupled with an attention mechanism for sentiment analysis. To our knowledge, this is the first work to combine all these three models.
- The attention mechanism is proved to increase the model's performance to realize the sentiment analysis by capturing the words responsible for sentiment at any position in the sentence.
- We conduct comprehensive experiments on IMDb and SSTb datasets. Our model CAGRN outperformed state-of-the-art models with a few parameters.

The remainder of the paper is structured as follows. Section II discusses the related work to our sentiment analysis model. A detailed description of CAGRNarchitecture is provided in Section III. The experiment setup and results are described in Section IV. Finally, section V concludes the paper with a final remark.

2 Related Work

A large number of researchers have been interested in sentiment analysis. As a result, a variety of ways have been offered. In sentiment analysis, deep learning algorithms such as CNNs and RNNs and their modifications have shown superior outcomes. CNN, Bi-GRU, and the attention model are all used in our research. Therefore, this section discusses different proposed models related to our work.

2.1 Convolutional Neural Networks

A large number of CNNs based models have been proposed for sentiment analysis; Kim (2014) used a multi-channel CNN trained on top word2vec pre-trained word embedding. Y. Zhang et al. (2017) conducted the sensitivity analysis of CNN models to prove the effect of CNN architecture on the performance. The study (Xu et al., 2017) applied a deep CNN for multilingual sentiment analysis. Conneau et al. (2017)investigated the effectiveness of deeper CNN to deal with the long-range association of the sentence. Similarly, Johnson & Zhang (2017)applieddeep pyramid CNN for capturing long-range dependencies. Also, Johnson & Zhang (2014)also utilized a CNN-based model for the best use of word order in the text representation. Kalchbrenner et al. (2014) used a

CNN-based model network that handles varying length input sentences and captures short and long-range dependencies. Santos & Gatti (2014) proposed a deep CNN that exploits character-to-sentence-level features and detects negation. The study byX. Zhang et al. (2015) designed a CNN that proves the usefulness of character information in text classification.

2.2 Recurrent Neural Networks

Many RNN-based approaches have been proposed in the literature to learn sequence inputs and represent long-range dependencies. Similarly, Lin et al. (2018) applied the structure-attention LSTM to model the contextual information. Also,Mousa & Schuller (2017) explored a generative contextual Bi-LSTM to learn each word's right and left context in the sentence. Likewise, a gated RNN model was proposed by M. Zhang et al. (2016) to capture semantic and syntactic information as well as represent the context in which a word appears. Long-range dependencies, on the other hand, have been captured using RNNs.Yang et al. (2017)developed an LSTM model to deal with a long input sentence and a target aspect discriminative features.

Meanwhile, Mujika et al. (2017) applied a Fast-Slow RNN to long-range model dependencies and map complex features. To represent

the words of the phrase in the form of parent-child relationships in the tree structure, Taiet al. (2015)built a tree-structured LSTM approach. Likewise, N. Wang et al. (2017) designed a Bi-GRU model coupled with anattention mechanism to learn long-term dependencies

Also, Chen et al. (2017) introduced a multiple attention LSTM to learn the dependencies separated by a large distance. Kokkinos & Potamianos (2017)suggested an attention-based GRU with a tree structure model where the informative nodes are selected based on the weighted representation of the sentence. Yequan Wang et al. (2016)created an LSTM model with aspect embedding and an attention mechanism that learns aspects in the text at a long-range.

2.3 Hybrid Neural Networks

For sentiment analysis, there is currently a substantial number of hybrid models. Here are a few that are relevant to our work.Hassan & Mahmood (2017)built an approach that augments the CNN with the LSTM layer, which replaces the CNN's pooling layer. The study by Zhou et al. (2016) invented a model that integrates a Bi-LSTM and CNN model with two dimensions convolutional and two-dimensional max pooling. Nguyen & Nguyen (2017) constructed a model that extracts the local features using a

combination of the semantic rules from the lexicon and features produces by a Deep CNN. Afterward, the produced features are fed to the Bi-LSTM to generate the final representation of the sentence that helps capture long-term dependencies. The research by R. Zhang et al. (2016) suggested a dependency sensitivity CNN model that learns the hierarchical representation with LSTM. Then the CNN applies different filters to learn the features. Yenter & Verma (2017) applied a combination of several branches of deeper CNN-LSTM for sentiment analysis.

However, our proposed model CAGRNis different from the former in following points: the models byHassan & Mahmood (2017), R. Zhang et al.(2016), Yenter & Verma(2017) use the LSTM, and Nguyen & Nguyen(2017), Zhou et al.(2016) utilize the Bi-LSTM whereas CAGRN uses Bi-GRU for capturing long-term dependencies. Also, Zhou et al. (2016) approachalso use two-dimensional CNN, whereas our model uses one-dimensional CNN. Furthermore, our model applies multiple CNN with different filters to allow CAGRN to capture different local features while the former does not use. Another difference is that in (R. Zhang et al., 2016; Zhou et al., 2016), the CNN is built on top LSTM and Bi-GRU, whereas in CAGRN, the Bi-GRU learns the features from CNN. Lastly, the former does not use the attention mechanism, while the

CAGRNapplies it to select the sentiment's important words carefully.

In a nutshell, the existing approaches are more computationally expensive than our model and cannot represent the contextual information at the same level as our model does.

3 Proposed Method

This section discusses the problem definition and the details of theCAGRNmodelproposed to solve the problem. CAGRNarchitecture is shown in Fig.1. The CAGR Nconsists of five main parts: word embeddings layer, convolutional and max-pooling layer, Bi-GRU layer, attention layer, and Output layer.

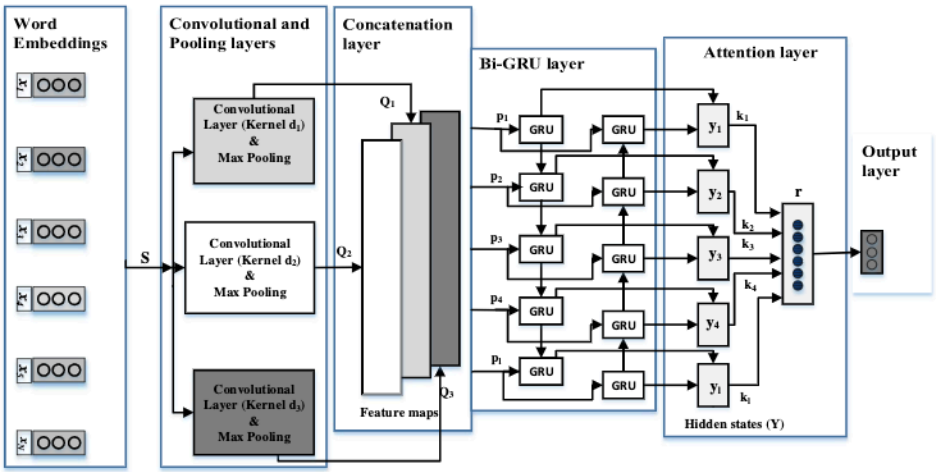

Figure 1: Architecture of Convolutional Attention Gated Recurrent Network.

The input to the model is a matrix $S = [x_1, x_2, x_3, x_4, x_5, x_N] \in \mathbb{R}^{d \times N}$ where $x_i \in \mathbb{R}^d$ is the word in the sentence, and N is the length of sentence S. We apply to the input three concurrent convolution operations with kernel size d_1, d_2 and d_3, respectively. Afterward, the max-pooling operations are applied to the final feature maps. The resulted feature maps are concatenated and fed to the Bi-GRU. The hidden states produced by the Bi-GRU are fed to the attention layer that produces the weighted representation of the sentence. Finally, the model applies the output layer to obtain the final prediction of the sentence

3.1 Problem Definition

Formally, in this work, we propose the contextual sentiment analysis defined as follows. Let us consider the input sentence S with length N, $S = [x_1, x_2, x_3, x_4, x_5, x_N] \in \mathbb{R}^{d \times N}$ where $x_i \in \mathbb{R}^d$ corresponds to the i^{th} word vector in the sentence matrix. The purpose of our approach is to give the sentiment label to each word x_i using the contextual information. We claim that additional words in the same sentence can be used to determine the polarity of a particular word x_i. S, i.e. $[x_j | \forall j \leq N, j \neq i]$, hold the key long-term dependencies and local contextual information necessary for sentiment analysis.

3.2 Word Embedding layer

In case there is no large supervised training set, one of the alternatives to improve the performance of the models is to use unsupervised neural language models to initialized the word vectors (Socher et al., 2011). In addition, Kim (2014) proved that using the unsupervised neural language models is a good ingredient in NLP, especially in sentiment analysis. Therefore, we used GloVe[1](Pennington et al., 2014)context predicting model, which is publicly available. GloVe word embedding inherits the benefits

[1]Available from: https://github.com/stanfordnlp/GloVe

offered by global matrix factorization and local context methods. GloVe has been trained on Wikipedia 2014 and Gigaword 5 with a total number of 6 billion tokens. During the training process, we fine-tuned the embeddings. This word embedding method allows our model todeal with important features like contextual, syntactic, and semantic features.

Let S be the sentence input to the model. After encoding, $S = [x_1, x_2, x_3, x_4, x_5, x_N] \in \mathbb{R}^{d \times N}$ where$x_i \in \mathbb{R}^d$ is the word in the sentence, d is the embedding dimension, and N is the length of sentence S.

3.3 Convolutional and Pooling layers

We propose three concurrent convolutional layers coupled by max-pooling layers, motivated by the effectiveness of CNN models presented in the literature.The length of the sentence often determines the structure of the convolutional that is applied to the sentence representation and the length of the word, which are denoted by N and d, respectively.The convolutional layer creates a feature map M by applying a filter with the weight matrix $F \in \mathbb{R}^{d \times n}$on a window of n words in the sentence matrix S. Equation (1) gives the ith element of the feature map M in formal terms:

$$M_i = \sigma(\sum(M[*, i: i + n] \odot F) + b) \quad (1)$$

where b is a bias term, and σ is a non-linear function, normally tanh or ReLu. $M[*, i: i+n]$ is from the i^{th} to $i+n^{th}$ word vectors in the sentence matrix. $\odot$ is the element-wise product between two matrices. The filter F is applied to each possible window of words in the sentence vector to generate the final feature map M given by equation (2):

$$M = [M_1, M_2, M_3, \dots, M_{N-n+1}], M \in \mathbb{R}^{N-n+1} \quad (2)$$

We apply max pooling to the resulting feature map described by equation 2 because of its performance in discovering critical features with minimal computational cost. We use size two max-pooling, which halves contiguous features in the feature map M by extracting the maximum among them.

The max-pooling operation transforms the feature map M to $Q \in \mathbb{R}^{\left\lfloor \frac{N-n+1}{2} \right\rfloor}$. Formally, Q is defined by:

$$Q = \left[Q_1, Q_2, Q_3, \dots, Q_{\left\lfloor \frac{N-n+1}{2} \right\rfloor}\right] \quad (3)$$

Therefore, stimulated by the idea (Kim, 2014), we apply multiple filters with sizes 4, 5, and 6 to get the final feature mapP which is the concatenation of individual feature maps Q_4, Q_5 and Q_6. Thus, P is given by the following equation:

$$P = [Q_4 \oplus Q_5 \oplus Q_6] \quad (4)$$

where $\oplus$denotes the concatenation operator.

Applying multiple kinds of filters with different sizes helps capture possible local contextual features over the sentence matrix S.

3.4 Bi-directional Gated Recurrent Unit

RNNs are the type of feed-forward neural network that is specialized in the modeling input sequence and long-range dependencies. In this work, we adopt the Bi-GRU variant of RNNs suggested to overcome the vanishing and exploding gradient (Hochreiter, 1998) that the traditional RNNs suffer. Bi-GRU learns the input in forward and backward directions. Modeling the input sequence in both directions allows the model to have previous and upcoming contextual information. Therefore, this solvesthe bias problem that single channel RNN suffers.

The input to our Bi-GRU is the encoded features produced by the CNN layer. We represent the encoded features by $P = \{p_1, p_2, p_3, \dots, p_l\}$ with length l. The Bi-GRU is made of forward GRU and backward GRU layers. The forward GRU outputs a sequence $\overrightarrow{H}$, a set of hidden vectors produced in the forward direction while the backward GRU produces a sequence$\overleftarrow{H}$, a set of

hidden vectors produced in the backward direction. Finally, the hidden vectors $\overrightarrow{H}$ and $\overleftarrow{H}$are concatenated to make the final output Y for the Bi-GRU.

The outputs for both forward and backward layers are calculated using the standard GRU updating equations described below. Formally, the GRU has different gates that govern the operations of the unit. At time step t, the GRU outputs the hidden vector h_t that is a linear interpolation of the previously hidden vector h_{t-1} and the candidate activation$\tilde{h}$. During this operation, the update gate z regulates how much the unit updates its activation while the reset gate r_t allows the unit to forget the previous computation and pretends that the input sequence starts.

To sum up, the computation process of GRU hidden unit j at time t is governed by the equation (5)-(8)(Cho et al., 2014):

$$h_t^j = (1 - z_t^j)h_{t-1}^j + z_t^j + \tilde{h}_t^j \quad (5)$$

$$z_t^j = \sigma(W_z p_t + U_z h_{t-1})^j \quad (6)$$

$$\tilde{h}_t^j = \tanh(W p_t + U_h(r_t \odot h_{t-1}))^j \quad (7)$$

$$r_t^j = \sigma(W_r p_t + U_r h_{t-1})^j \quad (8)$$

where σ is the sigmoid function, $\odot$is the element-wise multiplication, U_z, U_h, U_r are weight matrices.

The output of the GRU is a vector H containing hidden vectors $H= [h_1, h_2, h_3, \dots, h_l]$. Therefore, H is the output of the forward GRU layer, denoted as$\overrightarrow{H}$. While, the backward GRU does the same thing, except that its input sequence is reversed, thus, its output is denoted by $\overleftarrow{H}$.

The Bi-GRU layer generates an output vector
$Y = [y_1, y_2, y_3, \dots, y_l]$ in which each element $y_t$is a concatenation of the forward and backward hidden states.

$$y_t = [\overrightarrow{h_t} \oplus \overleftarrow{h_t}] \quad (9)$$

where $\oplus$signifies the element-wise sum of the two hidden state vectors, forward and backward.

3.5 Attention Mechanism

For sentiment classification, not all words in the phrase are equally relevant. Therefore, we propose an attention mechanism that helps to prioritize the important word responsible for sentiment in the sentence. This study applies a simplified attention mechanism applied in neural translation (Luong et al., 2015). The attention mechanism in our model works on each output $y_t$of the Bi-GRU.

With this attention mechanism, our model can model the long-term information at any location in the sentence.

Let $Y = [y_1, y_2, y_3, \ldots, y_l]$ be the input to the attention layer. Y is a matrix representing the output of Bi-GRU.

$$g_t = \tanh(W_m y_t + b_m) \quad (10)$$

$$k_t = \frac{\exp(g_t{}^T c_t)}{\sum_{t=1}^{l} \exp(g_t{}^T c_t)} \quad (11)$$

$$r = \sum_{t=1}^{l} k_t y_t \quad (12)$$

where l is the number of encoded-word features in the sentence, g_t is a hidden attention vector, k_t is a vector containing the normalized weight for an encoded feature of word x_i, c_t is global contextvector, and r is the weighted representation of the encoded features of the sentence.

3.6 Output Layer

The output layer gets the vector r, a weighted representation of a sentence's encoded features as input. After that, for each sentiment class label, the softmax is used to estimate the probability distribution. The softmax process is defined in precise terms as follows:

$$P(y_i = k|b_i; w_k) = \frac{e^{w_k^T b_i}}{\sum_{j=1}^{C} e^{w_j^T b_i}} (13)$$

where C is the number of classes, b_i and w_k are bias and weight for class k.

For each training sample, we use the cross-entropy loss to reduce the difference between the actual probability distribution and the anticipated probability:

$$L_i = -\sum_{k=1}^{C} t_k(y_i) \log P(y_i = k | b_i; w_k) \quad (14)$$

where $t_k(y_i)$ is a one-shot vector that represents the actual sentiment label distribution, $P(y_i = k|b_i; w_k)$ is the predicted probability.

Data	S	Train	Val	Test	C
IMDb	50000	37500	6250	6250	2
SSTb	11855	8544	1101	2210	2

Table 1: Statistics of the datasets used

S, C denote the number of samples and classes, respectively.

4 Experiments

This section presents the details of the datasets used to evaluate the effectiveness of the CAGRNmodel, CAGRNhyper-parameters, and training details. Finally, it shows the experimental results obtained by CAGRNand the comparison with baseline models.

4.1 Datasets

We evaluated the performance of our model on IMDb[2](Maas et al., 2011) Large Movie Review and SSTb[3](Socher et al., 2013)Stanford Sentiment Treebank datasets. We evaluate the model for binary sentiment classification. For the first dataset IMDb, samples are balanced 50% for training and 50%for testing. In addition, the reviews in this dataset contain multiple sentences. While evaluating our model, we used75% for training,12.5 % validation, and the remaining 12.5% for testing. We did not follow the proposed distribution because we wanted to give our model many training samples and provide a validation set. The second dataset, SSTb, consists of 11,855 movie reviews collected from the Rotten Tomatoes site. The reviews in the dataset contain a single short

[2]Available from: http://ai.stanford.edu/~amaas/data/sentiment/

[3]Available from: https://nlp.stanford.edu/sentiment/

sentence per review. The statistics for each dataset are presented in Table 1.

4.2 CAGRN Hyper parameters

The embeddings with dimension 200 that GloVe initialized are the inputs to the model. We utilize three channels in the CNN model, each with a one-dimensional convolutional layer with 256 filters and a kernel size of d(4,5,6). For each convolutional layer, we employ the rectified linear units (ReLu) activation function. In addition, each channel employs a two-size max-pooling layer. On the IMDb and SSTb datasets, the Bi-GRU uses the hidden state of size 300 and 70, respectively. On both datasets, the number of epochs used to train the proposed model differs between (3,8). We set the batch size to 32 for each iteration of the training procedure. To prevent our model from overfitting, we applied the early stopping and dropout (Srivastava et al., 2014). We applied the dropout probability between 0.5 and 0.8 after the convolution layer and after the Bi-GRU layer. The model was trained via Adam optimizer (Kingma &Ba, 2015) with default parameters. We minimized the cross-entropy loss given by equation 14 when training our model. Finally, we used the Keras Python package with the TensorFlow backend to create our model.

4.3 Baseline models

We compare the effectiveness of our proposed model to the following state-of-art approaches:

PL(Maas et al., 2011) is a probabilistic model that performs sentiment analysis. The IMDb dataset was designed in this work.

RNTN(Socher et al., 2013) is a well-known recursive neural tensor network that represents the sentence in the form of a tree. The SST dataset was created in these results.

CNN-multichannel(Kim, 2014) is a commonly used model that applies multiple convolutional with different filters to perform sentiment analysis.

DCNN (dynamic CNN) (Kalchbrenner et al., 2014)is a graph-based model of the features of a sentence.

DeepCNN(Santos & Gatti, 2014) uses character information and sentence representations for the sentiment classification.

CNN (Semantic CNN) (Yin et al., 2017)augments the features extracted by CNN with sentiment information from the lexicon.

CNN-SA (CNN Sensitivity Analysis)(Y. Zhang & Wallace, 2017)performs the analysis of effect of CNN architecture to the results in sentiment analysis.

	Model	IMDb	SSTb
Baseline models	CNN-multichannel	–	88.1
	DCNN		
	DeepCNN	–	86.8
	SCNN	–	85.7
	CNN-SA	–	87.9
	RNTN	–	85.49
	CNN-LSTM	–	85.4
	Tree-GRU	–	88.3
	BLSTM-2DCNN	–	89.5
	DAN	–	89.5
	DSCNN	89.4	86.3
	PL	90.7	89.1
	HRL	88.89	–
	CBA+LSTM	90.92	–
	Deep CNN-LSTM	90.1	–
		89.5	–
Ours	**CGRN**	91.39	89.71
	CAGRN	**91.50**	**89.83**

Table 2: *Results of our models and baseline models*

CNN-LSTM(Hassan & Mahmood, 2017)is a hybrid approach that uses an LSTM layer to replace CNN's pooling layer.

Tree-GRU(Kokkinos & Potamianos, 2017)represents the information in a sentence as a tree, with nodes picked according to the weight of each word.

BLSTM-2DCNN(Zhou et al., 2016) is a combined approach of BLSTM with two-dimensional CNN.

DAN (Deep Averaging Network)(Iyyer et al., 2015) is a simple but efficient model that performs the sentiment analysis by ignoring the syntactic structure of the inputs.

DSCNN (Dependency sensitivity CNN) (R. Zhang et al., 2016)uses an LSTM for sentences representation and applies CNN in extracting the features.

HRL(Yiren Wang & Tian, 2016)is a Hybrid Residual LSTM model that performs sequence classification by combining the ResNet connection with LSTM.

CBA+LSTM (Cognitive Based Attention LSTM)(Long et al., 2017) is an approach that represents the attention of a given the word in a sentence. Also, it captures the attention of a given sentence in the document.

Deep CNN-LSTM(Yenter & Verma, 2017) applies multiple branches of hybrids of CNN and LSTM.

4.4 Experimental Results

The evaluation results achieved by our models and baseline models are shown in Table 2. We report the results in terms of accuracy expressed in percentage. It is observed that our proposed model CAGRN obtained superior results compared to the state-of-the-art models.CAGRN improved theperformance of both datasets. CAGRN raised the accuracy by 0.58 on IMDb and 0.3-3 on SSTb.Among the results presented in the literature, the model with the highest accuracy is Hybrid Residual LSTM (HRL) (Yiren Wang & Tian, 2016) with 90.92% on IMDb and Tree-GRU (Kokkinos & Potamianos, 2017) with 89.5% on SSTb.

The good performance of our model is related to the advantages of using contextual information extracted by the combined approaches. In addition, the attention mechanism helps our model perform a wise selection of important words containing the sentiment information at any location in the sentence. The results reveal that our model, without attention, CGRN, got comparable results to the baseline models.

In brief, the experimental results strongly agree with our idea of using contextual information to perform the sentiment classification.

5 Conclusion

In this paper, we augment the CNN with a Bi-GRU joined with an attention mechanism to perform the contextual sentiment analysis. In particular, the multiple convolutional applied helps the model to extract possible local features by retaining the order of the input sentence. On the other hand, the Bi-GRU learns global features. Besides, the attention mechanism helps the model to select the important words responsible for the sentiment information. We evaluated the effectiveness of the proposed model for binary sentiment classification on IMDb and SSTb datasets. The obtained results reasonably agree with our idea of using contextual information to realize contextual sentiment analysis.

The experiments, including performing various ablation experiments of our model's components, will be done in future work. In addition, our model can be applied to other sequence learning tasks. Specifically, future work in neural machine translator can investigate whether this can enjoy the beauty of our model.

REFERENCES

Bahdanau, D., Cho, K., & Bengio, Y. (2015). *Neural Machine Translation by Jointly Learning to Align and Translate*. *In*. *ICLR*.

Bespalov, D., Bai, B., Shokoufandeh, A., & Qi, Y. (2011). *Sentiment Classification Based on Supervised Latent n-gram Analysis*. *In*. *CIKM*, 375–382.

Chen, P., Sun, Z., Bing, L., & Yang, W. (2017). *Recurrent Attention Network on Memory for Aspect Sentiment Analysis*. *In*. *EMNLP*, 452–461.

Cho, K., Merrienboer, B. Van, Bahdanau, D., & Bengio, Y. (2014). *On the Properties of Neural Machine Translation : Encoder-Decoder Approaches*. *In*. *SSST*, 103–111.

https://doi.org/10.3115/v1/W14-4012

Collobert, R., & Weston, J. (2008). *A unified architecture for natural language processing*. *In*. *ICML*, 160–167.

https://doi.org/10.1145/1390156.1390177

Conneau, A., Schwenk, H., Le Cun, Y., & Barrault, L. (2017). *Very Deep Convolutional Networks for Text Classification*.

In. EACL, *1*, 1107–1116. https://doi.org/10.1007/s13218-012-0198-z

Deng, L., & Yu, D. (2013). *Deep Learning: Methods and Applications*. *Foundations and Trends in Signal Processing*, *7*(3–4), 197–387.

https://doi.org/10.1561/2000000039

Hassan, A., & Mahmood, A. (2017). *Deep Learning approach for sentiment analysis of short texts*. *In. IEEE-ICCAR*, 705–710.

Hochreiter, S. (1998). *The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions*. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *06*(02), 107–116.

Hochreiter, S., & Schmidhuber, J. (1997). *Long Short-Term Memory*. *Neural Computation*, *9*(8), 1735–1780.

Iyyer, M., Manjunatha, V., Boyd-Graber, J., & III, H. D. (2015). *Deep Unordered Composition Rivals Syntactic Methods for Text Classification*. *In. ACL-IJCNLP*, *1*, 1681– 1691.

Johnson, R., & Zhang, T. (2014). *Effective Use of Word Order for*

Text Categorization with Convolutional Neural Networks. *In NAACL*, 103–112. http://arxiv.org/abs/1412.1058

Johnson, R., & Zhang, T. (2017). *Deep Pyramid Convolutional Neural Networks for Text Categorization*. *In*. *ACL*, 562–570.

Kalchbrenner, N., Grefenstette, E., & Blunsom, P. (2014). *A Convolutional Neural Network for Modelling Sentences*.*In*. *ACL*, 655–665.

Kim, Y. (2014). *Convolutional Neural Networks for Sentence Classification*. *In*. *EMNLP*, 1746–1751.

Kingma, D. P., & Ba, J. L. (2015). Adam: *A Method for Stochastic Optimization*. *ICLR*, 1–15.

Kokkinos, F., & Potamianos, A. (2017). *Structural Attention Neural Networks for improved sentiment analysis*. *In*. *ECACL*, *2*, 586–591.

Lin, K., Lin, D., & Cao, D. (2018). *Sentiment Analysis Model Based on Structure Attention Mechanism*. *In Advances in Computational Intelligence Systems*, *650*.

Liu, B. (2012). *Sentiment Analysis and Opinion Mining*. *Morgan &*

Claypool Publishers, (May), 1–108.

Long, Y., Qin, L., Xiang, R., Li, M., & Huang, C.-R. (2017). *A Cognition Based Attention Model for Sentiment Analysis*. *In*. *EMNLP*, 473–482.

Luong, M.-T., Pham, H., & Manning, C. D. (2015). *Effective Approaches to Attention-based Neural Machine Translation*. *In EMNLP*, 1412–1421.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011). *Learning Word Vectors for Sentiment Analysis*. *In*. *ACL*, 142–150.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). *Efficient Estimation of Word Representations in Vector Space*. *In*. *ICLR*.

Mousa, A. E.-D., & Schuller, B. (2017). *Contextual Bidirectional Long Short-Term Memory Recurrent Neural Network Language Models*: *A Generative Approach to Sentiment Analysis*. *In*. *EACL*, *1*, 1023–1032.

Muhammad, A., Wiratunga, N., & Lothian, R. (2016). *Contextual sentiment analysis for social media genres*. *Knowledge-*

Based Systems, *108*, 92–101.

Mujika, A., Meier, F., & Steger, A. (2017). *Fast-Slow Recurrent Neural Networks*. *In*. *NIPS*.

Nguyen, H., & Nguyen, M.-L. (2017). *A Deep Neural Architecture for Sentence-Level Sentiment Classification in Twitter Social Networking*. In *In*. *PACLING* (pp. 15–27).

https://doi.org/10.1016/B978-0-444-51747-0.50005-6

Pang, B., & Lee, L. (2005). *Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales*. *In*. *ACL*, (1), 115–124.

Pang, B., & Lee, L. (2008). *Opinion Mining and Sentiment Analysis*. *Foundations and Trends in Information*, *1*(2), 1–135.

Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up?: *sentiment classification using machine learning techniques*. *In*. *EMNLP*, *10*, 79–86.

Pennington, J., Socher, R., & Manning, C. (2014). *Glove: Global Vectors for Word Representation*. *In*. *EMNLP*, 1532–1543.

Poria, S., Cambria, E., & Gelbukh, A. (2016). *Aspect extraction for*

opinion mining with a deep convolutional neural network. *Knowledge-Based Systems*, *108*, 42–49.

Pozzi, F. A., Fersini, E., Messina, E., & Liu, B. (2017). *Challenges of Sentiment Analysis in Social Networks: An Overview*. In *Social analysis in social netwiorks* (Vol. 1, pp. 1–11). Elsevier Inc. https://doi.org/10.1016/B978-0-12-804412-4.00001-2

Santos, C. N. dos, & Gatti, M. (2014). *Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts*. *In*. *COLING*, 69–78.

Socher, R., Pennington, J., Huang, E. H., Ng, A. Y., & Manning, C. D. (2011). *Semi-Supervised Recursive Autoencoders for Predicting Sentiment Distributions*. *In EMNLP*, 151–161.

Socher, R., Perelygin, A., & Wu, J. (2013). *Recursive deep models for semantic compositionality over a sentiment treebank*. *In*. *EMNLP*, 1631–1642.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: *A Simple Way to Prevent Neural Networks from Overfitting*. *Journal of Machine Learning Research*, *15*, 1929–1958.

Taboada, M., Brooke, J., Tofiloski, M., Voll, K., & Stede, M. (2011). *Lexicon-Based Methods for Sentiment Analysis*. *Computational Linguistics*, *37*(2), 267–307.

Tai, K. S., Socher, R., & Manning, C. D. (2015). *Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks*. *In*. *ACL*.

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*. *IJCNLP*, 174–179.

Wang, S., & Manning, C. (2012). *Baselines and Bigrams: Simple, Good Sentiment and Topic Classification*. *In*. *ACL*, (July), 90–94.

Wang, Y., Huang, M., Zhao, L., & Zhu, X. (2016). *Attention-based LSTM for Aspect-level Sentiment Classification*. *In*. *EMNLP*, 606–615.

Wang, Y., & Tian, F. (2016). *Recurrent Residual Learning for Sequence Classification*. *In*. *EMNLP*, 938–943.

Wang, Z., & Zhang, Y. (2017). *Opinion Recommendation using*

Neural Memory Model. *In*. *EMNLP*, 1626–1637.

Wilson, T., Wiebe, J., & Hoffmann, P. (2005). *Recognizing contextual polarity in phrase-level sentiment analysis*. *In HLT/EMNLP*, (October), 347–354.

Xu, L., Lin, J., Wang, L., Yin, C., & Wang, J. (2017). *Deep Convolutional Neural Network based Approach for Aspect-based Sentiment Analysis*. *In*. *ASTL*, *143*(Ast), 199–204.

Yang, M., Tu, W., Wang, J., Xu, F., & Chen, X. (2017). *Attention-Based LSTM for Target-Dependent Sentiment Classification*. *In*. *AAAI*, 5013–5014.

Yenter, A., & Verma, A. (2017). *Deep CNN-LSTM with combined kernels from multiple branches for IMDb review sentiment analysis*. *IEEE-UEMCON*, 540–546.

Yin, R., Li, P., & Wang, B. (2017). *Sentiment Lexical-Augmented Convolutional Neural Networks for Sentiment Analysis*. *In*. *IEEE-DSC*, 630–635.

Zhang, L., Wang, S., & Liu, B. (2018). *Deep Learning for Sentiment Analysis*: A Survey. *Wiley*

Interdiscip. *Rev*. *Data Min*. *Knowl*. *Discov*., *vol*. *8, no*. *4, 2018*.

Zhang, M., Zhang, Y., & Vo, D. (2016). *Gated Neural Networks for Targeted Sentiment Analysis*. *In*. *AAAI*, 3087–3093.

Zhang, R., Lee, H., & Radev, D. (2016). *Dependency Sensitive Convolutional Neural Networks for Modeling Sentences and Documents*. *In*. *NAACL-HLT*, 1512–1521.

Zhang, X., Zhao, J., & LeCun, Y. (2015). *Character-level Convolutional Networks for Text Classification*. *In*. *NIPS 2015*, 1–9

Zhang, Y., & Wallace, B. (2017). *A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification*. *In*. *IJCNLP*, 253–263.

Zhou, P., Qi, Z., Zheng, S., Xu, J., Bao, H., & Xu, B. (2016). *Text Classification Improved by Integrating Bidirectional LSTM with Two-dimensional Max Pooling*. In. COLING, 2(1), 3485–3495.