Entire Information Attentive GRU for Text Representation

Guoxiu He, Wei Lu*
School of Information Management, Wuhan University
Wuhan, Hubei, China
{guoxiu.he,weilu}@whu.edu.cn

ABSTRACT

Recurrent Neural Networks (RNNs), such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), have been widely utilized in sequence representation. However, RNNs neglect variational information and long-term dependency. In this paper, we propose a new neural network structure for extracting a comprehension sequence embedding by handling the entire representation of the sequence. Unlike previous works that put attention mechanism after all steps of GRU, we add the entire representation to the input of the GRU which means the GRU model takes the entire information of the sequence into consideration in every step. We provide three various strategies to adding the entire information which are the Convolutional Neural Network (CNN) based attentive GRU (CBAG), the GRU inner attentive GRU (GIAG) and the pre-trained GRU inner attentive GRU (Pre-GIAG). To evaluate our proposed methods, we conduct extensive experiments on a benchmark sentiment classification dataset. Our experimental results show that our models outperform state-of-the-art baselines significantly.

CCS CONCEPTS

• **Information systems → Document representation**; **Sentiment analysis**; • **Computing methodologies → Neural networks**; *Supervised learning by classification*;

KEYWORDS

Document Representation; Sentiment Analysis; Gated Recurrent Unit

ACM Reference Format:
Guoxiu He, Wei Lu. 2018. Entire Information Attentive GRU for Text Representation. In *2018 ACM SIGIR International Conference on the Theory of Information Retrieval (ICTIR '18), September 14–17, 2018, Tianjin, China.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3234944.3234947

1 INTRODUCTION

Capturing semantics of texts is a challenge of finding machine understandable representation for nature language understanding. Based on researches which aim to obtain semantically meaningful distributed representation of individual words, such as word embedding [2, 16], much remains to be done to obtain satisfying representations of sequences. Deep neural networks (DNN) have gained great popularity to encode distributed vector representation of a sequence.

The deep learning methods are divided into two categories. The one is trained by unsupervised learning [7] such as the Paragraph Vectors [12], SkipThought vectors [10], Document Vector through Corruption (Doc2VecC) [3] and so on. The other consists of models trained specifically for a certain task [14] which depends on downstream applications and trained by supervised learning. Several models have been proposed such as Recurrent Neural Networks (RNNs) [5, 8] and Convolutional Neural Networks (CNNs) [9]. The specifically trained sequence embedding performs better than generic ones. For one task, CNN based models can only encode spatial information instead of sequential information. For RNN based models, the most common way is to add a max or average pooling across all time steps [13], or just pick up the hidden representation at the last time step as the encoded embedding. However, the last step of RNN always forgets some information and every step of RNN only takes the information before into consideration but the entire contextual information for all words in the sequence.

Recent years, attention mechanism proposed to use on the top of CNN and RNN based models has shown great success in many Natural Language Processing (NLP) tasks such as machine translation [1], question answering system [22] and recognizing textual entailment [18] which introduces extra source of information to guide the extraction of sequence embedding [6]. For example, attention based RNN models [1], each time-step hidden representation is weighted by attention distribution. However, the previous attention mechanism can't solve the problem that RNNs have. To tackle this problem, we encode the entire contextual information of the sequence by attention mechanism into every step of RNN.

In this paper, we study the problem mentioned above for extracting sequence embedding. The contributions are as follows: (1) We present a novel structure to encode the entire contextual information into every step of RNN model. (2) We propose three various strategies to enhance the GRU model: the CNN based attentive GRU (CBAG), the GRU inner attentive GRU (GIAG) and the pre-trained GRU inner attentive GRU (Pre-GIAG). (3) We evaluate our models on a benchmark sentiment classification dataset named IMDB and achieve new state-of-the-art results.

2 THE PROPOSED MODEL

We encode the contextual information into every step of GRU using the entire representation attentive GRU (ERAG) structure with three types of attention mechanisms, which are the CNN based

*The corresponding author.

ICTIR '18, September 14–17, 2018, Tianjin, China

ACM ISBN 978-1-4503-5656-5/18/09…$15.00
https://doi.org/10.1145/3234944.3234947

attentive GRU (CBAG), the GRU inner attentive GRU (GIAG) and pre-trained GRU inner attentive GRU (Pre-GIAG).

2.1 Overview

Each example of the task is represented as a set (S, y), where $S = (w_1, w_2, \cdots, w_M)$ is a sequence with a length M, $y \in Y$ is the label representing the category of S. We can represent the task as estimating the conditional probability $Pr(y|S)$ based on the training set, and predicting the category for testing examples by $y^* = argmax_{y \in Y} Pr(y|S)$.

In this section, we propose an Entire Representation Attentive GRU (ERAG) structure to estimate the probability distribution $Pr(y|S)$ through the following layers.

Word Embedding Layer: The goal of this layer is to represent $i-$th word w_i in S with a d dimensional vector $\mathbf{x}_i \in \mathbb{R}^d$. The word embedding is a fixed vector for each individual word, which is pre-trained by GloVe or Word2Vec.

Entire Representation Layer: The purpose of this layer is to get the entire contextual information of the sequence. We utilize a common CNN model or a RNN model to encode all embeddings $\mathbf{X} \in \mathbb{R}^{M \times d}$ of sequence S. The output is $\mathbf{c} = Whole\_Representation(\mathbf{X})$, where $\mathbf{c} \in \mathbb{R}^h$.

Entire Representation Attentive GRU Layer: In this model, we put the entire semantic representation $\mathbf{c}$ into input of the GRU as an attention mechanism to take the contextual information into consideration for every step. Then we aggregate the sequence into a fix-length vector using the Bi-GRU model. In this work, we implement GRU[4] which is parameterized as fellows:

$$\mathbf{z} = \sigma(\mathbf{W}_Z \mathbf{x}_t + \mathbf{U}_Z \mathbf{h}_{t-1})$$
$$r = \sigma(\mathbf{W}_T \mathbf{x}_t + \mathbf{U}_T \mathbf{h}_{t-1})$$
$$\mathbf{s} = tanh(\mathbf{W}_S \mathbf{x}_t + \mathbf{U}_S(\mathbf{h}_{t-1} \circ \mathbf{r}))$$
$$\mathbf{h}_t = (1 - \mathbf{z}) \circ \mathbf{s} + \mathbf{z} \circ \mathbf{h}_{t-1}$$

Prediction Layer: To this end, we employ an one layer feed-forward neural network to consume the fix-length vector, and apply the *sigmoid* or $softmax$ function in the output layer to get the expected label.

2.2 CNN Based Attentive GRU (CBAG)

As attention mechanism aims at finding useful and important part of a sequence, the first model applies the above intuition directly. Instead of using the original words embedding to the RNN model and putting attention mechanism above all steps of RNN model, we weight the embedding of every word according to the contextual representation encoded by CNN model.

First, we get a contextual representation $\mathbf{c}$ of the sequence based on a CNN model.

$$\mathbf{c} = CNN(\mathbf{X})$$

Then, the relevance, which is named attention in this paper, between $t-$th step input $\mathbf{x}_t$ and the contextual representation $\mathbf{c}$ is calculated as follows:

$$\alpha_t = \sigma(\mathbf{c}\mathbf{M}\mathbf{x}_t)$$

where $\mathbf{M} \in \mathbb{R}^{h \times d}$ is a convert matrix.

Finally, the $\tilde{\mathbf{x}}$ which is $\mathbf{x}_t$ weighted by α_t is the input of the GRU.

$$\tilde{\mathbf{x}}_t = \alpha_t * \mathbf{x}_t$$

In this way, every word representation is encoded by introducing the representation of whole sequence, which means all steps of RNN are considering contextual information of the sequence. The CBAG model is shown in Figure1.

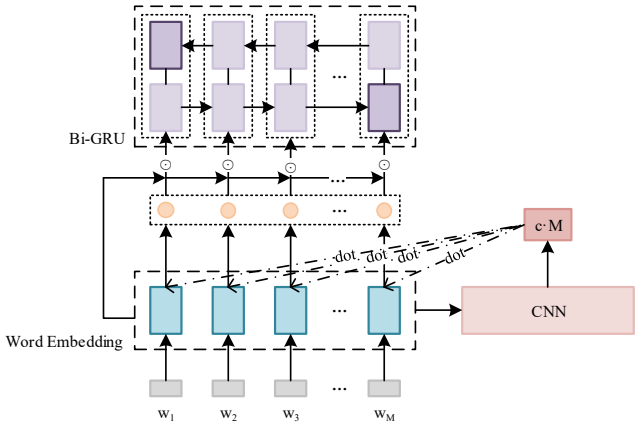

Figure 1: The CNN based attentive GRU model which weights the input of GRU with attention mechanism.

2.3 GRU Inner Attentive GRU (GIAG)

Inspired by the previous work of LSTM [8] on solving the gradient exploding problem in RNN, instead of adding contextual information to the original input, we can apply attention deeper to the GRU inner activation. Because these inner activation units control the flow of the information within the hidden stage and enable information to pass long distance in a sequence [22], we add the entire contextual information to these active gates to influence the hidden representation as follows:

$$\mathbf{z} = \sigma(\mathbf{W}_Z \mathbf{x}_t + \mathbf{U}_Z \mathbf{h}_{t-1} + \mathbf{M}_Z \mathbf{c})$$
$$\mathbf{r} = \sigma(\mathbf{W}_T \mathbf{x}_t + \mathbf{U}_T \mathbf{h}_{t-1} + \mathbf{M}_T \mathbf{c})$$
$$\mathbf{s} = tanh(\mathbf{W}_S \mathbf{x}_t + \mathbf{U}_S(\mathbf{h}_{t-1} \circ \mathbf{r}))$$
$$\mathbf{h}_t = (1 - \mathbf{z}) \circ \mathbf{s} + \mathbf{z} \circ \mathbf{h}_{t-1}$$

Where $\mathbf{M}_Z$ and $\mathbf{M}_T$ are attention weight matrices. In this way, the update and forget units in GRU can focus on not only long and short term memory but also the representation of the entire sequence as contextual information. The architecture of GIAG is shown in Figure2.

The structure of the Pre-trained GRU inner attentive GRU(Pre-GIAG) is nearly same as the GIAG model. The main difference is that the contextual information of whole sequence is the pre-trained representation encoded by the GRU model. And before putting the representation into the Entire Representation Attentive GRU Layer, we convert the vector into a more suitable matrix space for the second GRU model. We convert the vector with one dense layer shown bellow:

$$c' = Relu(\mathbf{M}_{convert}\mathbf{c})$$

Where $\mathbf{M}_{convert} \in \mathbb{R}^{h \times h}$ is the weight matrix.

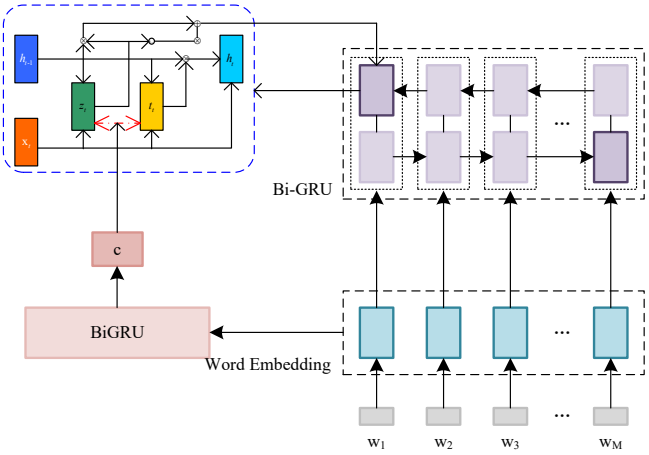

Figure 2: The GRU Inner Attentive GRU which adds the entire sequence representation encoded by the first GRU as contextual information into the whole representation attentive GRU layer.

3 EXPERIMENT

We evaluate our Entire Representation Attentive GRU (ERAG) structure on sentiment analysis task using IMDB dataset and compare it with state-of-the-art sentence embedding methods.

3.1 Dataset

The IMDB movie review dataset contains 50,000 movies reviews categorized as either positive or negative. It comes with predefined train/test split [15]: 25,000 reviews are used for training and 25,000 for testing. The two classes are balanced in the training and testing sets. The result of this dataset is one of benchmarks for sentiment analysis task.

3.2 Baselines

We compare against the following sentence representation baselines:

bag-of-words (BoW) [23]: A simple and the most commonly representation of sentence.

RNN-LM [17]: A recurrent neural network based language model.

Denoising Autoencoders (DEA) [21]: A representation learned from reconstructing original sentence x using corrupted one $\hat{x}$.

Word2Vec+IDF and Word2Vec+AVG [16]: A representation generated through weighted average of word vectors learned using Word2Vec.

Doc2Vec [12]: The main idea of Doc2Vec is a target word is predicted by the word embeddings of its neighbors in together with a unique document vector learned for each document.

Skip-thought Vectors [10]: A generic, distributed sentence encoder that extends the Word2Vec skip-gram model to sentence level.

Document Vector through Corruption (Doc2VecC) [3]: It represents each document as a simple average of the word embeddings of all the words in the document. It is the state-of-the-art model.

Table 1: The result on the the IMDB dataset compared our proposed models with all baseline models

Model	Error rate %
Bag-of-Words (BOW)	12.53
Denoising Autoencoders (DEA)	11.58
Word2Vec + AVG	12.11
Word2Vec + IDF	11.28
Paragraph Vectors	10.81
Skip-thought Vectors	17.42
Doc2VecC	10.48
RNN-LM	13.59
LSTM with tuning and dropout	13.50
LSTM initialized with word2vec	10.00
CNN based attentive GRU	9.82
GRU inner attentive GRU	10.06
pre-trained GRU inner attentive GRU	9.86

3.3 Settings

We remove the stop words using NLTK for the whole dataset. Performances are chosen to be classification error. For our new structure, we using Glove as pre-trained word embeddings which are set not trainable during training.

For **CBAG**, we use CNN to encode the whole contextual information of the sequence. In this CNN model, we just use one Convolutional1D layer followed by one GlobalMaxPooling1D layer, one dropout layer, one dense layer, one dropout layer and one BatchNormalization layer. The Convolutional1D layer is set with 250 filters and 3 kernels. The dropout rate is set as 0.5. And the hidden dimensions of dense layers in CNN model and attention layer are all set as 300.

For **GIAG and Pre-GIAG**, we use GRU model to encode the whole information of the sequence. We set the hidden dimensions of all steps, the weight matrix of the contextual information vector, the second GRU model and followed dense layer are all set with 300 dimensions.

4 RESULT AND ANALYSIS

To demonstrate the effectiveness of the proposed structure which contains three different types of attention mechanisms, we compare our method with all baselines including state-of-the-art methods. The results are shown as Table1.

On the one hand, the structure we proposed yields a significant performance gain compared to BoW method. Compared to RNN-LM, the attention mechanism of the ERAG structure leads to better performance. Because the ERAG structure is trained by supervised learning, it can outperform many unsupervised learning methods such as Word2Vec+IDF/AVG and DEA. In addition, all three attention mechanism based models can take contextual information into consideration during training the GRU model, so it can beat all baseline models.

On the other hand, because the CBAG model combines the advantages of CNN which encodes the spacial information of whole sequence as contextual knowledge and RNN which encodes the sequential information of a sequence, the result of CBAG is better

than the GIAG model which is just modeled by RNN. In addition, the error rate of the Pre-GIAG model is lower than the GIAG model because using the pre-trained model to get the contextual information trained by supervised learning can encode the entire representation better.

In summary, the structure that encoding the contextual information of entire sequence by attention mechanism into the GRU model is better than others.

5 RELATED WORKS

5.1 Sentence Embedding

There are many works to extract sentence representation divided into two mian streams. On the one hand, the model with the goal of self-prediction inspired by auto-encoder idea makes the embedding of a sentence useful for predicting the sentence itself. For example, ParagraphVector [12] predict the words in sentences to get the sentence embedding. Sequential Denoising Auto-encoders (SDAE) [7] applies a LSTM-based encoder-decoder architecture [4] which takes the word orders into consideration by predicts the original sentences by decoder.

On the other hand, the model extracts sentence embedding trained by predicting adjacent sentence such as SkipThought [10] and FastSent [7]. A special case for this line of work is sentence embedding learned by predicting bilingual parallel sentences with the encoder-decoder architecture [4], which was evaluated in [7].

Apart from unsupervised sentence embedding models, the supervised ways construct the embedding either by max/average pooling on the word representations in sentences, or by taking the end hidden state from a vanilla LSTM, which depend on the downstream application.

5.2 Attention Model

Many recent works have shown that the performance of deep neural networks can be improved by attention mechanism.

In attention based models, one representation is built with attention from other representation. For example, Memory Networks [24] uses an external memory to store the knowledge and the memory are read and written on the fly with respect to the attention, and these attentive memory are combined for inference. Since then, many variants have been proposed to extract semantically representation of sentence to solve all kinds of NLP tasks [11, 19].

Some works [20] try to introduce the attention mechanism into the LSTM-RNN architecture. RNN models the input sequence word-by-word and updates its hidden variable recurrently. In attention based RNN models, after computing each time step hidden representation, attention information is added to weight each hidden representation, then the hidden states are combined with respect to that weight to obtain the sematically representation.

6 CONCLUSIONS

We present a new neural network structure that consists attention mechanism applied to GRU model. Our results show that all these three models are able to outperform state-of-the-art model relying on taking the entire contextual representation into consideration for every step of RNN model.

7 ACKNOWLEDGEMENTS

We would like to thank National Demonstration Center for Experimental Library and Information Science Education, Wuhan University for sharing the GPU devices. This work is partially supported by the Major projects of the National Social Science Fund under Grant No.17ZDA292.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).
[2] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *JMLR* 3, Feb (2003), 1137–1155.
[3] Minmin Chen. 2017. Efficient vector representation for documents through corruption. *arXiv preprint arXiv:1707.02377* (2017).
[4] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).
[5] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).
[6] Cıcero Nogueira dos Santos, Ming Tan, Bing Xiang, and Bowen Zhou. 2016. Attentive pooling networks. *CoRR, abs/1602.03609* (2016).
[7] Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning distributed representations of sentences from unlabelled data. *arXiv preprint arXiv:1602.03483* (2016).
[8] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.
[9] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188* (2014).
[10] Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *NIPS*. 3294–3302.
[11] Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. 2016. Ask me anything: Dynamic memory networks for natural language processing. In *ICML*. 1378–1387.
[12] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *ICML*. 1188–1196.
[13] Ji Young Lee and Franck Dernoncourt. 2016. Sequential short-text classification with recurrent and convolutional neural networks. *arXiv preprint arXiv:1603.03827* (2016).
[14] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130* (2017).
[15] Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *ACL*. 142–150.
[16] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).
[17] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model.. In *Interspeech*, Vol. 2. 3.
[18] Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiskỳ, and Phil Blunsom. 2015. Reasoning about entailment with neural attention. *arXiv preprint arXiv:1509.06664* (2015).
[19] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *NIPS*. 2440–2448.
[20] Ming Tan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2015. LSTM-based deep learning models for non-factoid answer selection. *arXiv preprint arXiv:1511.04108* (2015).
[21] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *ICML*. ACM, 1096–1103.
[22] Bingning Wang, Kang Liu, and Jun Zhao. 2016. Inner attention based recurrent neural networks for answer selection. In *ACL*.
[23] Sida Wang and Christopher D Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *ACL*. 90–94.
[24] Jason Weston, Sumit Chopra, and Antoine Bordes. 2014. Memory networks. *arXiv preprint arXiv:1410.3916* (2014).