International Journal of Computational Intelligence Systems
Vol. 12(2), 2019, *pp.* 723–732
DOI: https://doi.org/10.2991/ijcis.d.190710.001; ISSN: 1875-6891; eISSN: 1875-6883
https://www.atlantis-press.com/journals/ijcis/

International Journal of Computational Intelligence Systems

Attention Pooling-Based Bidirectional Gated Recurrent Units Model for Sentimental Classification

Dejun Zhang[1,*], Mingbo Hong[2], Lu Zou[2], Fei Han[2], Fazhi He[3], Zhigang Tu[4], Yafeng Ren[5]

[1] *Faculty of Information Engineering, China University of Geosciences, Wuhan, Hubei 430074, China*
[2] *College of Information and Engineering, Sichuan Agricultural University, Yaan, Sichuan 625014, China*
[3] *School of Computer, Wuhan University, Wuhan, Hubei 430072, China*
[4] *School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 637553, Singapore*
[5] *Collaborative Innovation Center for Language Research & Service, Guangdong University of Foreign Studies, Guangzhou, Guangdong 510420, China*

ARTICLE INFO

Article History

Received 08 May 2018
Accepted 27 Jun 2019

Keywords

Natural language processing
Neural network
Gated recurrent units
Text classification

ABSTRACT

Recurrent neural network (RNN) is one of the most popular architectures for addressing variable sequence text, and it shows outstanding results in many natural language processing (NLP) tasks and remarkable performance in capturing long-term dependencies. Many models have achieved excellent results based on RNN. However, most of these models overlook the locations of the keywords in a sentence and the semantic connections in different directions. As a consequence, these methods do not make full use of the available information. Considering that different words in a sequence usually have different importance, in this paper, we propose bidirectional gated recurrent units (BGRUs) which integrates a novel attention pooling mechanism with max-pooling operation to force the model to pay attention to the keywords in a sentence and maintain the most meaningful information of the text automatically. The presented model allows to encode longer sequences. Thus, it not only prevents important information from being discarded but also can be used to filter noises. To avoid full exposure of content without any control, we add an output gate to the GRU, which is named as text unit. The proposed model was evaluated on multiple tasks, including sentimental classification, movie review data, and a subjective classification dataset. Experimental results show that our model can achieve excellent performance on these tasks.

1. INTRODUCTION

Recently, neural networks have achieved outstanding performance in many natural language processing (NLP) tasks, such as statistical machine translation [1], speech recognition [2,3], and natural language inference [4]. As one of the main parts of NLP, sentence modeling aims at representing the meaning of sentences. As a traditional method, the bag-of-words (BOWs) model addresses sentences as sets of words [5]; however, it usually suffers from the curse of dimensionality and a lack of rich meanings in the representation of words. In addition, many tasks require excellent sentence modeling to finish their works in NLP. The main challenge of sentence modeling is extracting the features from the words or *n*-grams and representing the sentence's meaning.

Recurrent neural network (RNN) can capture the long-term dependencies and learn the meaning of words. However, many RNN-based models face two problems: (1) They have ignored important information that may contribute to the understanding of the sentence. For example, the famous RNN-based model [6] utilized RNN to take the expression of the whole sentence into consideration rather than focused on the influence of rich semantic words, resulting in a lack of semantic and emotional emphasis on each word of the sentence. (2) They have failed to consider the differences between semantic expressions in different directions [7]. However, in the language of expression, the meaning of the sentence is relevant to its context. As a consequence, different directions of the sentence are important for sentence understanding.

To address these problems, Lai, Xu *et al.* [8] proposed recurrent convolutional neural networks (CNNs) for text classification. In this work, the Bidirectional RNN (BRNN) was employed to capture the semantic information in different directions and it integrated the max-pooling operation with CNN to extract the important information automatically. As their experimental results demonstrated, this approach could filter out noises effectively. Nevertheless, this model disposed the representation of the sentence obtained by BRNN and the CNN was employed to capture the important information of sentence instead of evaluating and selecting the words that expresses semantic information exactly.

In [9], an attention mechanism was proposed. The representation of a sentence was accumulated in each time dimension. Therefore, inspired by hierarchical attention networks for document classification [9], when we try to understand what a sentence means, the

[*] *Corresponding author. Email: zhangdejun@cug.edu.cn*

words contribute more to the sentence expression will facilitate us to better understand the sentence. Figure 1 illustrates the examples of meaningful words those can help us pay more attention to the sentence. In the first sentence, the words *sweet, funny, charming*, and *delightful* deliver strongly positive meanings. By contrast, *far-flung*, *illogical*, and *plain stupid* deliver negative meanings in the second sentence. Compared with other words such as *it* and *and*, they are more meaningful. The last sentence contains both the words expressing positive and negative meanings, but the negative meaning of the expression is more significant. Different from [9], our model is dedicated to maintaining more important information and reducing the effects of noise, rather than summing of all the noise and important information to express the meaning of the sentence.

Hence, our proposed attention pooling evaluates importance weights by representing sentences that are obtained by bidirectional gated recurrent units (BGRUs) [1]. In addition, we utilize the GRU to effectively overcome the situation where the gradient of RNN tends to vanish. Then, we can obtain the importance weight of each word and combine the corresponding weights with the representations of sentences to increase the weights of the important words in the time-step and feature dimension. However, since we retain all the information on the time-step and feature dimension, there will still be some noise information. Therefore, we propose the attention pooling mechanism combined with 2D max-pooling operation to consider the feature and time-step dimensions at the same time, which allows us to discover more valuable information.

In summary, in this paper we propose BGRU with an attention pooling. Compared with RNN, BGRU can process semantic relations in different directions of texts, which allows us to deal with more sequences of semantic information. We also incorporate attention pooling to assist in the extraction of information and determine which information is more important and needs to be preserved. For convenience, we call our model BGRU-Att-pooling. In our experiments, we evaluate our model based on sentimental classification [10], the Subjectivity dataset [11], and the Movie Review (MR) dataset [12]. Our evaluations show that our model can achieve a good result compared to a wide range of baseline models.

To sum up, our contributions are as follows: (1) We consider the locations of the keywords in a sentence and the semantic connections in different directions for sentimental classification. (2) Based on max-pooling mechanism, we propose an attention pooling mechanism which allows us to prevent more important information from being discarded but also can be used to filter noise. (3) Compare GRU to long short-term memory (LSTM), a text unit is added to GRU, which effectively controls the content gate rather than completely exposing it without any control on the stream information.

This paper is organized in several parts. Section 2 introduces the related work about text classification. In Section 3, we illustrate our BGRU-Att-pooling model and its implementation details are described. In Section 4, we introduce the details of our experimental evaluation setup. In Section 5, we present the experimental results. Finally, the conclusions and future work are discussed in Section 6.

2. RELATED WORK

As one of the mainstream tasks of NLP, text classification plays an important role and has great research and application value. In particular, the cost of manually processing massive data is prohibitive. Excellent performance has been achieved on sentimental classification [13–15] and spam categorization [16,17], which belong to the field of text classification.

In the early stages, Tong *et al.* [18] proposed SVMs for text classification, which can reduce the need for a labeled training set. Due to the prominent performance of neural networks, many neural network models [19–21] have been proposed for text classification. Unlike conventional approach, which relied heavily on human knowledge, neural networks can completely extract features automatically. Sinha *et al.* [19] proposed an end-to-end neural network-based model for hierarchical classification. They utilized external knowledge in the form of topic category taxonomies to facilitate the classification by introducing an adapted version of attention to represent documents dynamically through the hierarchy. Wang *et al.* [20] proposed a label-word joint embedding method. Different from the methods just utilize labels as the supervision, it embeds the words and labels in the same joint space, and measures the compatibility of word-label pairs to attend the document representations. Qian *et al.* [21] proposed linguistically regularized LSTM, which was trained with linguistic resources. The linguistic knowledge was applied to the classification in order to combine the features of humans with the neural network through the sentimental lexicons to assist with sentimental analysis.

Our model is also aimed at training sentence expressions and classifying them; however, neural networks are expected to help us automatically rather than utilizing the sentimental lexicons. Lee *et al.* [22] proposed sequential short-text classification with recurrent and CNNs, which utilized RNN and CNN to generate word embeddings, and then classified the short texts through a fully connected layer and a softmax layer. However, this operation results in high time consumption due to the training of word embeddings. As a contrary, in this work, we aim at presenting an approach that can automatically encode sentences, which enables the neural network to automatically perform sentimental classification and avoid complicated feature engineering process. Therefore, we utilize the pretrained word embeddings to replace the text instead of training the word representations of the text separately as employed in [22].

Briefly speaking, neural networks for sequence processing models can be broadly divided into two categories: recursive sentence

it's *sweet* $^{(+)}$, *funny* $^{(+)}$, *charming* $^{(+)}$, and completely *delightful* $^{(+)}$.

the story is *far-flung* $^{(-)}$, *illogical* $^{(-)}$, and *plain* $^{(-)}$ *stupid* $^{(-)}$.

a feel-*good* $^{(+)}$ picture in the *best* $^{(+)}$ sense of the term.

a *boring* $^{(-)}$ masquerade ball where normally *good* $^{(+)}$ actors, even kingsley, are made to look *bad* $^{(-)}$.

Figure 1 | Examples from the movie review dataset.

modeling, such as Recursive neural networks (RecNN) and recurrent sentence modeling, such as RNN, recursive sentence modeling defines recursive tree structures to express longer phrases [10], which combines the leaf nodes of the tree structures to represent compositionality. Tai *et al.* [23] proposed tree-structured LSTMs, which utilizes tree-structured network topologies to build the sentence model. Recurrent sentence modeling benefits from the recurrent structure of RNN, has achieved remarkable performance in capturing long-term dependencies. It addresses the variable-length sequence with the memory's state, in which each time-step's output depends on the value at the previous time. Recently, Hochreiter *et al.* [24] proposed LSTM units, it stores the previous state and memorizes the extracted features from the current time's input thus it consumes less time than RecNN. GRU is similar to LSTM of which each unit remembers the features of the input stream, which is beneficial for processing the long sentences that contain both important information and noise. Chung *et al.* [25] compared the performance between LSTM and GRU. From their conclusions, LSTM unit computes the new memory content without any control of the previous time step, and GRU exposes its full content without any control. Therefore, we finally package the content gate in the GRU to avoid its full exposure and we call its output as text unit. We analyze the impact of LSTM and GRU on our task in Section 5.

As one of the mainstream sentence modeling approaches, CNN also has outstanding performance. Kim [13] proposed training on pretrained word embeddings for sentimental classification, which utilized the convolutional layer to capture the features among the sentences and applied a max-over-time pooling operation [26] over the feature map and kept the maximum value. Mou *et al.* [27] observed that CNN can extract the word's features effectively while RNN performs well in capturing the inherent sentence structures. Based on this observation, they proposed TBCNN, which employs a CNN that is based on a tree structure. Johnson *et al.* [28] proposed a deep pyramid CNNs, they utilized the downsampling to decrease computational cost while efficiently representing long range associations in text. Li *et al.* [29] employ initializing convolutional filters with features that computed by K-means rather than initializing filters randomly, which enable features to be efficiently captured. Zhang *et al.* [30] proposed a grouped weight sharing way to instead of word embeddings. They supposed words derived from an external resource, which can be divided into N group. And this is similar to Li's parameter initialization through clustering [29]. Our model is consistent with its purpose.

For these CNN guided sentence models [13,26–30], the performance is limited by the size of the perception field, thus it is less effective in capturing the context of the sentence than the RNNs. In this work, one insight is that the implicit semantics of the sentence is often highly context-dependent, thus we utilize RNN to encode sentences which can better capture the context relationship, so that the model pays more attention to the most important information in the sentence. Both [29] and [30] employ cluster method to make their model more effective in extracting features, and obtain more discriminative sentence representation. As a contrast, we focus on reducing the complexity of model preprocessing during the training stage and make the features more significant which can be extracted more efficiently in the meanwhile.

As the main part of language modeling, learning a distributed representation for words and combining the representation with words in a sentence to deliver the information are challenging. Bengio *et al.* [31] proposed a neural probabilistic language model which overcame the curse of dimensionality and learned the distribution of words. In [32], Skip-gram and Continuous Bag-of-Words (CBOW) were proposed for computing representations of words. The Skip-gram model was used to predict each context word based on the current word. In contrast, the CBOW model predicts the current word based on the context word. Since then, Mikolov *et al.* [33] also showed that negative sampling can accelerate the process and learn more regular word representations. The complex Huffman binary tree was replaced with negative sampling to improve the training speed and enhance the quality of the word vectors. After that, Pennington *et al.* [34] proposed global vectors for word representation. In their work, the model was trained in a word–word co-occurrence matrix. Unlike in the previous model, the co-occurrence probabilities of words can reflect the correlation between the words, and this relationship can be described by the word–word co-occurrence matrix.

Through these models, we can obtain deeper word embeddings representations. For example, Vec("*King*") − Vec("*Queen*") = Vec("*man*") − Vec ("*women*"). In many NLP tasks, we would obtain the whole representation of a sentence by replacing each word with word embeddings. Furthermore, we achieve excellent performance in our task. Our evaluation is based on replacing the words with the pretrained word embeddings, which effectively helps the model focus on the keywords.

3. THE PRESENTED MODEL

3.1. Model Description

As shown in Figure 2, BGRU-Att-pooling consists of input layer (Section 3.2), BGRU (Section 3.3), attention pooling (Section 3.4), and output layer (Section 3.5). The model utilizes GRU to capture the long-term dependencies instead of employing CNN to learn the transformation in one shot, thus it encodes the representation of sentence progressively via recurrent process. The model did not directly utilize the output of BGRU as the final sentence representation for text classification since the direct output of BGRU without any post-processing contains much noise, and it will introduce more learnable parameters to the output layer. Instead, an adapted attention mechanism is applied to enhance the feature representation and the results are then fed into a max-pooling layer to retain valuable feature information. In practice, the feature enhancement operation can prevent hidden features that are helpful for classification but not captured by BGRU from being filtered out.

We show the model training procedure as follows. (1) Given a sentence X, it is first processed by word segmentation and each word is replaced with a pre-trained word embedding. (2) The resulted embedding matrix is employed as the input of BGRU, and its forward and backward representations are obtained through BGRU. (3) An improved attention pooling operation is applied to get words' weight values, the word with weight value larger than a threshold is recorded as keyword. (4) In order to reduce the computational complexity caused by the high-dimensional word embeddings, max-pooling is followed to make the sentence representation more semantic. (5) The sentence representation is flattened and classified by the softmax layer. Through the output layer, the model outputs the predicted label $\hat{y}$.

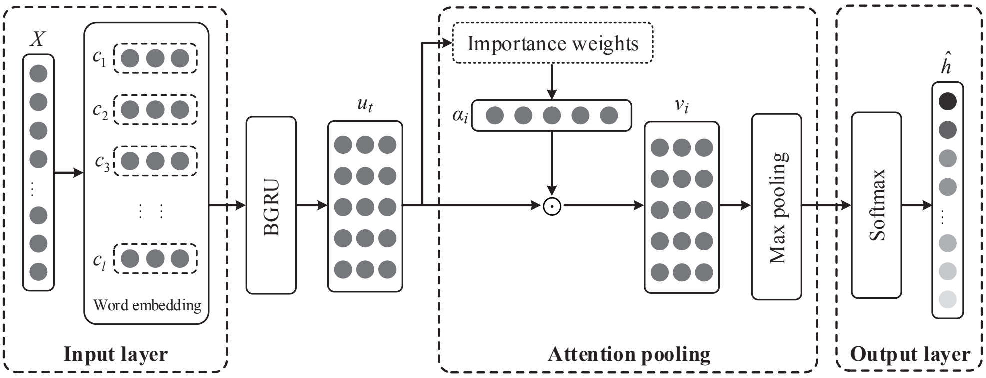

Figure 2 Architecture of the attention pooling-based bidirectional gated recurrent unit (BGRU).

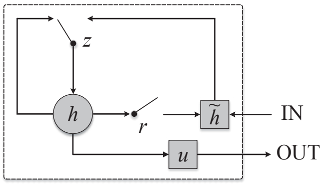

Figure 3 Illustration of the calculation process of gated recurrent unit (GRU).

3.2. Input Layer

For the proposed model, the input is a sentence. Each word in the sentence is replaced by the corresponding word vector, and it then forms a word embedding. Without loss of generality, let x_i denotes the i-th word in the sentence and X represents the input sentence. Let $c_i \in \mathbb{R}^d$ be the d-dimensional word vectors for the word x_i, and $C \in \mathbb{R}^{l \times d}$ represents the word-embedding matrix, where l is the max length of the sentence. The specific setting of the max length of the sentence and the way to initialize the word vector are described in Section 4.3.

3.3. Bidirectional Gated Recurrent Unit

In [1], a new activation function for RNN was proposed, which was called GRU and employed two gating units, namely, the reset and update gates, to control the stream of information. In addition, we add a text unit to control the stream of output.

Figure 3 shows the GRU with text unit. The GRU consists of reset gates r, update gates z, activation h, candidate activation $\tilde{h}$, and text unit u. After variable sequences pass through the hidden units, the output sequence is the representation of a sentence or phrase. The activation value h_t^j at time t is expressed as follows:

$$h_t^j = \left(1 - z_t^j\right) h_{t-1}^j + z_t^j \tilde{h}_t^j. \tag{1}$$

Equation (1) shows that the activation value h_t^j was determined by the update gate z_t^j, the previous state h_{t-1}^j, and the candidate state $\tilde{h}_t^j$. The update gate z_t^j determines how much of the stream of information can be kept or forgotten at time-steps t and t-1. The update gate z_t^j is expressed as follows:

$$z_t^j = \sigma\left(W_z c_t + U_z h_{t-1}\right), \tag{2}$$

where σ is a logistic sigmoid function, x_t is a vector of the sequences at time t, and W_z and U_z are weights that can be trained to update z_t^j. The candidate state is $\tilde{h}_t^j$:

$$\tilde{h}_t^j = \tanh\left(W_h c_t + U_h\left(r_t \odot h_{t-1}\right)\right)^j, \tag{3}$$

where $\odot$ is element-wise multiplication, r_t is the reset gate. W_h and U_h can be trained to contribute to the candidate state. The reset gate controls the previous state h_{t-1} by deciding how much previous information can be kept. W_r and U_r are weights that can be trained to update r_t. If the reset gate is zero, it means $\tilde{h}_t^j$ forgets all previous states, thus the reset gate will be updated as follows:

$$r_t = \sigma\left(W_r c_t + U_r h_{t-1}\right). \tag{4}$$

In this paper, we utilize GRU to capture the previous and later information, as shown in Figure 2. Our model contains forward and backward sequences that represent semantic relations in different directions. To capture the relationship between the forward and backward sequences, we utilize element-wise summation to combine the forward and backward sequences as follows:

$$h_i = h_i^F \oplus h_i^B, \tag{5}$$

where h_i^F is the forward sequence and h_i^B is the backward sequence.

$$u_t = \tanh\left(W_w h_i + b_w\right), \tag{6}$$

where u_t is text unit, rather than utilizing exposed content directly as text embedding.

Benefitting from GRU control of the information flow, GRU retains the previous information and combines the information that is currently entered, which allows us to obtain a better semantic representation.

3.4. Attention Pooling

Since different words have different meanings, we employ importance weights to identify important words that convey the meaning of the sentence. The formula is defined as follows:

$$\alpha_i = \frac{\exp\left(u_t^T u_w\right)}{\sum_t \exp\left(u_t^T u_w\right)}, \tag{7}$$

where u_w is used to assist the softmax function in automatically deciding which words play an important role. When the normalized importance weights are obtained, the most commonly used approach to get the final expression of the sentence for attention-based models is the weighted sum of all the word vectors at time step [9,35,36], its definition is as follows:

$$v_i = \sum_{i=0}^{t} a_i h_i. \tag{8}$$

However, directly superpose information at time step will lose some important information, hence, in order to retain more semantic information, we utilize element-wise multiplication instead. Therefore, the expression of the sentence is expressed as follows:

$$v_i = \alpha_i \odot h_i, \tag{9}$$

where $\odot$ refers to element-wise multiplication. Once the importance weight is close to zero, the multidimensional features of v_i are also close to zero.

Similarly, we also find that the retention of information will also bring about noise information. Therefore, we combine our attention mechanism with max-pooling to allow the model to encode longer sequences. In this way, the model not only prevents important information from being discarded but also can be used to filter noise. The operation about applying max-pooling to the matrix $v_{i,t}$ can be expressed as

$$O_{it} = Max\left(v_{i:i+k_1,t:t+k_2}\right). \tag{10}$$

where Max is the max-pooling function. We employ the filter $m^{k_1 * k_2}$ (k is the size of the filter) to extract the maximum in a fixed window, and the output h is expressed as

$$h = \left[O_{1,1}, O_{1,1+k_2} \ldots, O_{(l-k_1+1)/k_1,(d-k_2+1)/k_2}\right] \tag{11}$$

where l is the time-step length, and d is the size of the word embeddings.

3.5. Output Layer

The output of the max-pooling layer is the penultimate output of BGRU-Att-pooling. For text classification, we classify sentences into multiple classes that depend on various tasks. Then, we utilize the softmax function to predict the type of input text. We compute the result $\hat{y}$ as follows:

$$\hat{y} = softmax\left(Wh + b\right). \tag{12}$$

Then, the cross-entropy loss is defined as

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^{m} t_i \log\left(\hat{y}\right) + \lambda \|\theta\|_F^2, \tag{13}$$

where $t_i \in T^m$ is the one-hot representation of the true label, $\hat{y}$ is the result of the cross-entropy loss, and λ is an L_2 regularization [37] hyperparameter. We employ stochastic gradient descent and adopt the optimizer Adadelta [38].

4. EXPERIMENTAL SETUP

4.1. Experimental Settings

For demonstrating the efficiency and efficacy of our model, we conduct five experiments on a desktop machine with an Intel Core I7-6300 CPU (3.4 GHz) ans a GeForce 1080 GPU (16GB memory). We implemented our model based on Tensorflow framework and the source code is available at https://github.com/djzgroup/BGRU-AttentionPooling.

At the training stage, a one layer BGRUs is employed as our BGRU model and the number of hidden units is set as 300. We set model hyperparameter for the filter size, L_2 regularization rate, batch size, number of hidden units, dropout rate, and learning rate. For the dropout rate, the default value as 0.5 is set for all tasks. For the learning rate, the Adadelta optimizer is utilized which can be adapted dynamically on the basis of a single initial learning rate. Hence, we do not need to tune the learning rate during training. We also set the learning rate to a default value of 1.0 in all tasks, and the tanh function is chosen as the activation function according to the experimental results. Furthermore, the mini batch size is 10 and the L_2 regularization rate is 1×10^{-5}.

4.2. Datasets

In this paper, we evaluate our model on the following data sets, and Table 1 shows the summary statistics for these datasets.

- *SST*-1. The Stanford sentimental treebank[1] was proposed by [10], and it includes fine-grained labels (very negative, negative, neutral, positive, and very positive). SST-1 contains 8,544 sentences for training, 1,101 for validating, and 2,210 for testing. The data were provided in subsentence format. We train our model on both phrases and sentences in the training set and only test on sentences.
- *SST*-2. Similar to the SST-1; however, SST-2 eliminates neutral labels, integrates negative and negative labels into negative labels, and integrates active and active labels into positive labels.
- *MR*. The MR[2] dataset is a dataset with 5,331 positive and 5,331 negative reviews, proposed by [12].

Table 1 Summary statistics for the datasets.

Data	C	L	N	V	|Vpre|	Test
SST-1	5	18	11855	17834	17370	2210
SST-2	2	19	9613	16186	15802	1821
MR	2	21	10662	19472	17260	CV
Subj	2	23	10000	22240	19787	CV

C: number of target classes.
L: average sentence length.
N: number of sentence.
V: vocabulary size.
|Vpre|: number of words that are present in both the pretrained word embeddings and the dataset.
Test: test dataset size. (CV: These datasets do not set the test data. Hence, we utilize 10-fold cross validation to evaluate our model.)

[1] http://nlp.stanford.edu/sentiment/

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

- *Subj*. The subjective classification dataset (Subj))[2] was proposed by [11]. Subj includes 5,000 subjective sentences and 5,000 objective sentences.

4.3. Padding and Word Embeddings

4.3.1. Padding

Since we employ stochastic gradient descent, if we were to feed multiple batches, our model would require each sentence to be of a fixed length. The maximum sentence length for each dataset is denoted as *maxlen*, and sentences of length less than *maxlen* are padded.

4.3.2. Word embeddings

Our experiment utilizes GloVe[3] embeddings and employs the pretrained word embeddings, whose corpora were Wikipedia 2014 and Gigaword 5. We employ word embeddings of dimension 300. If a word appears in the pretrained word embeddings, it will be replaced with a word embedding. Otherwise, it will be randomly initialized using a uniform distribution in the range of [−0.1, 0.1]. Besides, the word embeddings are fine-tuned along with other trainable parameters.

4.4. Regularization and Dropout

We utilize the L_2 regularization [37] and dropout [39] to alleviate model overfitting. The dropout operation is expressed as

$$r_j^{(l)} \sim Bernoulli\left(p\right), \tag{14}$$

$$\tilde{y}^{(l)} = r^{(l)} * y^{(l)}. \tag{15}$$

By Equation (14), we obtain a vector that is generated by a Bernoulli process with probability p, where * denotes the element-wise product operation and the thinned output $\tilde{y}^{(l)}$ is sampled by $y^{(l)}$.

We employ the dropout operation in the word embeddings layer, the BGRU layer, and the Output layer. For the word embeddings layer, we apply the dropout operation after the word has been replaced by pretrained word embeddings. For the BGRU layer, we apply this operation in each unit. For the Output layer, we apply it before the softmax operation. We only perform L_2 regularization on the Output layer.

5. RESULTS

5.1. Performance

We show the results of evaluating our model and compare our two model variations (BGRU-Att and BGRU-Att-pooling) with other state-of-the-art models. The comparison results are shown in Table 2 and the best performing results are highlighted in bold.

Other models are summarized as follows: *MV-RNN*: Semantic compositionality through recursive matrix-vector spaces [40]; *RNTN*: Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank [10]; *DCNN*: A CNNs for modeling sentences [41]; *CNN-nonstatic/multichannel*: CNNs for Sentence Classification [13]; *Modeling-CNN*: Molding CNNs for text: nonlinear, nonconsecutive convolutions [42]; *Tree-LSTM/Dependency Tree-LSTM*: improved Semantic Representations From Tree-Structured LSTMs [23]; *C-LSTM*: A C-LSTM N [7]; *DSCNN*: Dependency Sensitive CNNs for Modeling Sentences and Documents [43]; *Li-Bi-BLSTM*: Linguistically Regularized LSTM for Sentiment Classification [21]; *d-TBCNN*: Discriminative Neural Sentence Modeling by Tree-Based Convolution [27]; *MVCNN*: Multichannel Variable-Size Convolution for Sentence Classification [44]; *RCNN*: Recurrent CNNs for Text Classification [8].

Table 2 Comparison of different models.

Model	SST-1	SST-2	Subj	MR
MV-RNN [40]	44.4	82.9	-	-
RNTN [10]	45.7	85.4	-	-
DCNN [41]	48.5	86.8	-	-
CNN-nonstatic [13]	48.0	87.2	93.4	81.5
CNN-multichannel [13]	47.4	88.1	93.2	81.1
Modeling-CNN [42]	51.2	88.6	-	-
Tree-LSTM [23]	51.0	88.0	-	-
RCNN [8]	47.2	-	-	-
C-LSTM [7]	49.2	87.8	-	-
d-TBCNN [27]	**51.4**	87.9	-	-
Dependency Tree-LSTM [23]	48.4	85.7	-	-
DSCNN [43]	49.7	89.1	93.2	81.5
Li-Bi-BLSTM [21]	48.6	-	-	82.1
MVCNN [44]	49.6	**89.4**	93.9	-
CNN-nonstatic + UNI [29]	50.8	89.0	93.7	82.1
BGRU-Att	49.8	88.4	93.4	81.4
BGRU-Att-pooling	49.7	89.2	**94.2**	**82.3**

SST-1 is the sentiment treebank's fined-grained classification.
SST-2 is the sentiment treebank's binary classification.
Subj is the subjective classification dataset's binary classification.
MR is the MR data's binary classification.
"-" indicates the model was not evaluated on this dataset.

For SST-1, d-TBCNN [27] has achieved the best result of 51.4%, and that for SST-2 is 89.4% (MVCNN [44]). Our model, namely, BGRU-Att-pooling, achieves outstanding results on MR and Subj, with test accuracies of 82.3% and 94.2%, respectively. Besides, our model still achieves comparable results on SST-1 and SST-1 for 49.7% and 89.2 %, respectively. From the results, we also found there is overfitting phenomenon on the SST datasets. As a consequence, we fail to achieve the best results on all datasets, and this is the problem we aim to resolve in the future. Besides, compare the proposed BGRU-Att-pooling to our implemented BGRU-Att, which discards the max-pooling operation, the results of our BGRU-Att-pooling model are superior on all datasets which demonstrates the effectiveness of max-pooling operation.

5.2. Analysis

5.2.1. Effects of filter size and training ratio

We are interested in evaluating the learning capability of our model. Thus, we divide the Subj dataset into train sets and test sets, with a training ratio for those datasets that do not specify a test set. As shown in Figure 4, it can be found that the model can achieve high accuracy while the training ratio is relatively small, and in the case of a small training set, our model still performs well in terms of learning ability. Besides, although there are some fluctuations as the training ratio increases, the overall performance of our model still increases.

[3] https://nlp.stanford.edu/projects/glove/

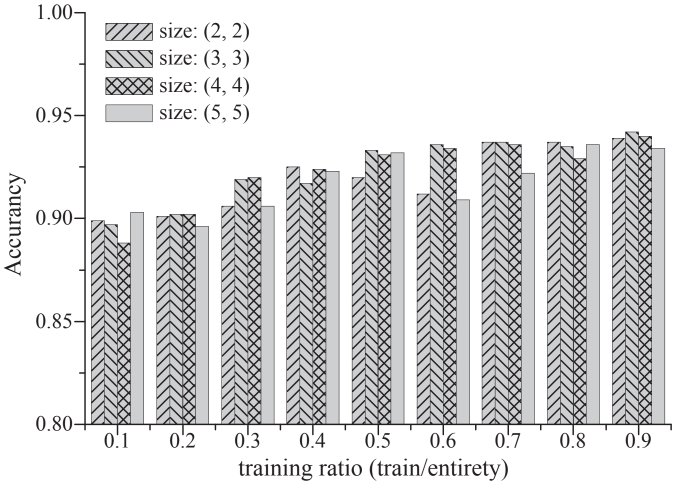

Figure 4 Illustration of the effects of different filter sizes and different training ratios on the experimental performance. The horizontal axis is the training ratio, which is the ratio of the training dataset size of to the size of the entire dataset.

We also investigate the influence of filter size on the experimental result. In this situation, each word's importance weight is calculated by Equation (7), and the importance weights are further combined with the representations of words according to Equation (9). The filter size affects how much information we will ultimately retain. For example, if we choose the a filter size of (5, 5), only the largest of the 25 elements will be retained.

As for max-pooling, larger pooling filter size results in smaller feature maps, and smaller feature maps retain more semantic information with the cost of losing more content information. In this experiment, we analyze the effect of different filter sizes on the experimental results. As shown in Figure 4, we found that when the filter size is chosen as (3, 3), we can get its optimal performance.

5.2.2. Dimension effect of word embeddings

We select pretrained word embeddings, whose corpora are Wikipedia 2014 and Gigaword 5, which also has 50, 100, and 200 dimension word embeddings. Figure 5 shows our model's performance with different word embeddings sizes.

From Figure 5, with the size of the word embeddings increases, the cost of the model increases, and the expression of each word is becoming more extensive. We found that our model performs best on the highest-dimensional word embeddings.

Thus, we can get a conclusion that different word vector sizes has different impact on the experimental results. According to the results, we recommend word embedding size = 300 in order to retain maximum feature information. In addition, we utilize max-pooling to alleviate the huge computation cost caused by the high-dimensional word vector.

5.2.3. Visualization of attention pooling

To evaluate the performance of the attention pooling in our tasks, we isolate the attention pooling for training and visualize the weights of the attention pooling (Equation (7)). In addition, we select a few sentences for data visualization analysis, as shown in Figure 6. The color's depth represents the importance weight's scale. The darker the color, the larger the importance weight is. The depth of the color represents the importance of the corresponding word in the sentence.

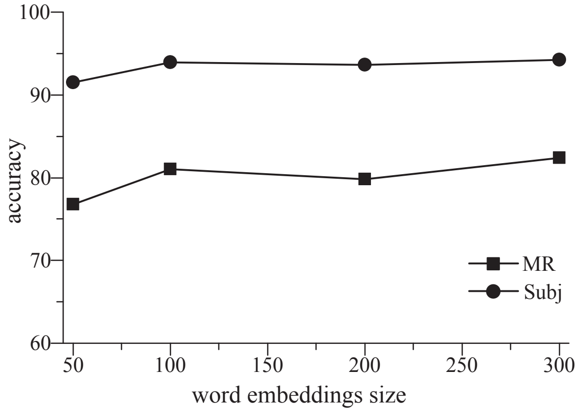

Figure 5 Illustration of the effects of word embeddings dimension.

In Figure 6, the first sentence is a positive sentence. The importance weights of "*funny*" and "*digressions*" are larger than those of the other words. Comparing "*funny*," "*digressions*," the color of the former is darker, which indicates that it is more important than the latter. By contrast, in the last sentence in Figure 6, the color of "*mediocre*" is deeper than those of other words, meaning that it has more negative semantics than the other words.

From Figure 6, the attention pooling is capable of paying attention to more representative words. Moreover, in the penultimate sentence in Figure 6, "*vicious*," "*messy*," "*uncouth*," and "*incomprehensible*" should be completely captured by attention pooling. However, in our experiments, "*uncouth*" and "*incomprehensible*" are not well captured. Our model still needs to be improved so that it captures more details.

5.2.4. Effect of sentence length and a comparison of LSTM and GRU

Our model combined with the improved attention pooling has the ability to filter noise and can capture more semantic information, which also makes it possible to encode longer sentence sequences for decoding.

To explore the effects of the length of sentences on the model performance, we conducted experiments using sentences with the max length and the average length on the Subj dataset, respectively.

As shown in Figure 7, the initial loss values of sentences with different lengths are very close. However, after a certain number of iterations, sentences with the max length converge significantly faster than sentences with the average length. This experiment proves that, due to the attention pooling, our model can handle more retained information and filter noise information to optimize model training. Besides, the GRU-based network converges faster than the LSTM-based network during the training process.

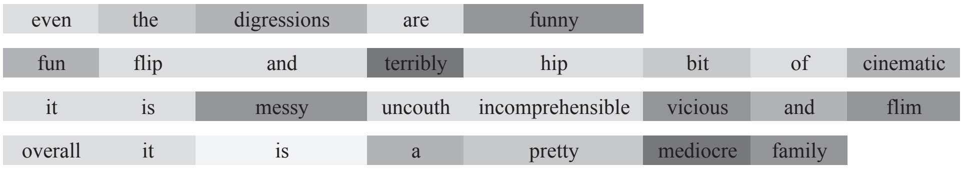

Figure 6 Illustration of visualization of the attention pooling.

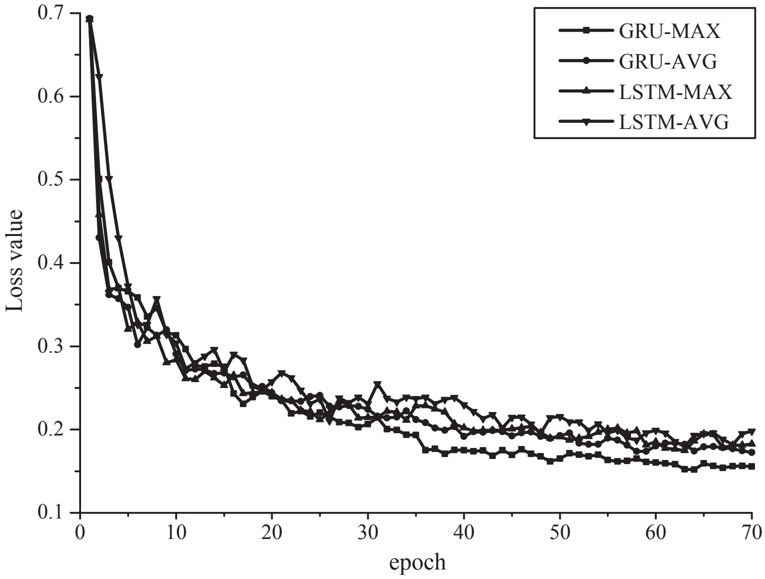

Figure 7 Illustration of the effects of sentence length and a comparison of long short-term memory (LSTM) and gated recurrent unit (GRU).

6. CONCLUSION

In this paper, we propose an attention pooling-based bidirectional GRU, namely BGRU-Att-pooling. With the help of a novel attention pooling mechanism, BGRU takes full account of the locations of the keywords in a sentence and the semantic connections in different directions thus it focuses on the important information in a sentence. In order to make the keywords more prominent in a sentence, we utilize max-pooling to retain the important information. Futhermore, a text unit is added to control the GRU's content gate to avoid completely exposing it. We discover that attention mechanism combined with the max-pooling operation outperforms the attention mechanism does not. Experiments demonstrate that our model can effectively extract useful information.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHORS' CONTRIBUTIONS

Dejun Zhang and Mingbo Hong proposed and implemented the algorithm, finished and analysed experiments together; Meanwhile, Dejun Zhang, Lu Zou and Fei Han wrote the draft manuscript. Mingbo Hong provided the datasets and proposed the idea of the comparison experimental design. Dejun Zhang and Yafeng Ren proposed the research issue, supervised the research by providing suggestions and revised the manuscript. Zhigang Tu and Fazhi He provided the experimental equipment.

Funding Statement

This work was supported by the National Natural Science Foundation of China under Grant 61702350, Grant 61472289 and Grant 61702121.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China under Grant 61702350, Grant 61472289, and Grant 61702121.

REFERENCES

[1] K. Cho, B. Van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, in Proceeding of the Conference on Empirical Methods Natural Language Process, Doha, 2014, pp. 1724–1734.
[2] G.E. Dahl, D. Yu, L. Deng, A. Acero, Context-dependent pretrained deep neural networks for large-vocabulary speech recognition, IEEE Trans. Audio Speech Lang. Process. 20 (2012), 30–42.
[3] E. Battenberg, J. Chen, R. Child, A. Coates, Y.G.Y. Li, H. Liu, S. Satheesh, A. Sriram, Z. Zhu, Exploring neural transducers for

end-to-end speech recognition, in Automatic Speech Recognition and Understanding Workshop, Okinawa, 2018, pp. 206–213.

[4] S. Lawrence, C.L. Giles, S. Fong, Natural language grammatical inference with recurrent neural networks, Knowledge Data Eng. IEEE Trans. 12 (2015), 126–140.

[5] S. Wang, C.D. Manning, Baselines and bigrams: simple, good sentiment and topic classification, in Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers, Association for Computational Linguistics, Jeju, 2012, vol. 2, pp. 90–94. https://dl.acm.org/citation.cfm?id=2390688

[6] T. Mikolov, M. Karafit, L. Burget, J. Cernock, S. Khudanpur, Recurrent neural network based language model, in INTERSPEECH 2010, Conference of the International Speech Communication Association, Chiba, 2010, pp. 1045–1048. https://www.isca-speech.org/archive/interspeech_2010/i10_1045.html

[7] C. Zhou, C. Sun, Z. Liu, F.C.M. Lau, A C-LSTM neural network for text classification, Comput. Sci. 1 (2015), 39–44. https://arxiv.org/abs/1511.08630

[8] S. Lai, L. Xu, K. Liu, J. Zhao, Recurrent convolutional neural networks for text classification, AAAI. 333 (2015), 2267–2273. https://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/viewPaper/9745

[9] Z. Yang, D. Yang, C. Dyer, X. He, A.J. Smola, E.H. Hovy, Hierarchical attention networks for document classification, in Proceeding of HLT-NAACL, San Diego, 2016, pp. 1480–1489.

[10] R. Socher, A. Perelygin, J. Wu, J. Chuang, C.D. Manning, A. Ng, C. Potts, Recursive deep models for semantic compositionality over a sentiment Treebank, in Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Seattle, 2013, pp. 1631–1642. https://aclweb.org/anthology/papers/D/D13/D13-1170/

[11] B. Pang, L. Lee, A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts, in Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, Barcelona, 2004, pp. 271–280.

[12] B. Pang, L. Lee, Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales, in Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, Ann Arbor, 2005, pp. 115–124.

[13] Y. Kim, Convolutional neural networks for sentence classification, in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Doha, 2014, pp. 1746–1751. http://www.aclweb.org/anthology/D14-1181

[14] J. Wehrmann, W. Becker, R.C. Barros, A multi-task neural network for multilingual sentiment classification and language detection on twitter, in The ACM/SIGAPP Symposium on Applied Computing, Pau, 2018.

[15] D. Zhang, F. He, S. Han, L. Zou, Y. Wu, Y. Chen, An efficient approach to directly compute the exact Hausdorff distance for 3D point sets, Integr. Comput. Aided Eng. 24 (2017), 261–277.

[16] H. Drucker, D. Wu, V.N. Vapnik, Support vector machines for spam categorization, IEEE Trans. Neural Netw. 10 (1999), 1048–1054.

[17] X. Wang, K. Liu, J. Zhao, Handling cold-start problem in review spam detection by jointly embedding texts and behaviors, in Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Long Papers), Association for Computational Linguistics, Vancouver, 2017, vol. 1, pp. 366–376. http://aclweb.org/anthology/P17-1034

[18] S. Tong, D. Koller, Support vector machine active learning with applications to text classification, J. Mach. Learn. Res. 2 (2001), 45–66. http://www.jmlr.org/papers/v2/tong01a.html

[19] K. Sinha, Y. Dong, J. Cheung, D. Ruths, A hierarchical neural attention-based text classifier, in Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, 2018, pp. 817–823.

[20] G. Wang, C. Li, W. Wang, Y. Zhang, D. Shen, X. Zhang, R. Henao, L. Carin, Joint embedding of words and labels for text classification, in Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (LongPapers), Melbourne, 2018, vol. 1, pp. 2321–2331. https://www.aclweb.org/anthology/P18-1216

[21] Q. Qian, M. Huang, J. Lei, X. Zhu, Linguistically regularized LSTM for sentiment classification, in Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Long Papers), Vancouver, 2017, vol. 1, pp. 1679–1689.

[22] J.Y. Lee, F. Dernoncourt, Sequential short-text classification with recurrent and convolutional neural networks, in Proceedings of NAACL-HLT, San Diego, 2016, pp. 515–520.

[23] K.S. Tai, R. Socher, C.D. Manning, 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 (Long Papers), Beijing, 2015, vol. 1, pp. 1556–1566.

[24] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (1997), 1735–1780.

[25] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, in NIPS Deep Learning Workshop, Montreal, Canada, 2014. https://nyu-staging.pure.elsevier.com/en/publications/empirical-evaluation-of-gated-recurrent-neural-networks-on-sequen

[26] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, P. Kuksa, Natural language processing (almost) from scratch, J. Mach. Learn. Res. 12 (2011), 2493–2537. http://www.jmlr.org/papers/v12/collobert11a.html

[27] L. Mou, H. Peng, G. Li, Y. Xu, L. Zhang, Z. Jin, Discriminative neural sentence modeling by tree-based convolution, in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Lisbon, 2015, pp. 2315–2325. http://aclweb.org/anthology/D15-1279

[28] R. Johnson, T. Zhang, Deep pyramid convolutional neural networks for text categorization, in Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Long Papers), Association for Computational Linguistics, Vancouver, 2017, vol. 1, pp. 562–570. http://aclweb.org/anthology/P17-1052

[29] S. Li, Z. Zhao, T. Liu, R. Hu, X. Du, Initializing convolutional filters with semantic features for text classification, in Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Copenhagen, 2017, pp. 1884–1889. https://www.aclweb.org/anthology/D17-1201

[30] Y. Zhang, M. Lease, B.C. Wallace, Exploiting domain knowledge via grouped weight sharing with application to text categorization, in Proceedings of the 55th Annual Meeting of the

Association for Computational Linguistics (Short Papers), Association for Computational Linguistics, Vancouver, 2017, vol. 2, pp. 155–160. http://aclweb.org/anthology/P17-2024

[31] Y. Bengio, R. Ducharme, P. Vincent, C. Jauvin, A neural probabilistic language model, J. Mach. Learn. Res. 3 (2003), 1137–1155. http://www.jmlr.org/papers/v3/bengio03a

[32] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, In: ICLR: Proceeding of the International Conference on Learning Representations Workshop Track, Arizona, USA, 2013, pp. 1301–3781. https://arxiv.org/abs/1301.3781

[33] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in NIPS'13 Proceedings of the 26th International Conference on Neural Information Processing Systems, Lake Tahoe, 2013, vol. 2, pp. 3111–3119. http://papers.nips.cc/paper/5021-distributed-representations-of-words-andphrases

[34] J. Pennington, R. Socher, C. Manning, Glove: global vectors for word representation, in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, 2014, pp. 1532–1543.

[35] F. Yang, A. Mukherjee, E. Dragut, Satirical news detection and analysis using attention mechanism and linguistic features, in Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Copenhagen, 2017, pp. 1979–1989. https://www.aclweb.org/anthology/D17-1211

[36] F. Dong, Y. Zhang, J. Yang, Attention-based recurrent convolutional neural network for automatic essay scoring, in Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), Association for Computational Linguistics, Vancouver, 2017, pp. 153–162. http://aclweb.org/anthology/K17-1017

[37] G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R.R. Salakhutdinov, Improving neural networks by preventing co-adaptation of feature detectors, arXiv: 1207.0580, 2012. https://arxiv.org/abs/1207.0580

[38] M.D. Zeiler, Adadelta: an adaptive learning rate method, arXiv: 1212.5701, 2012. https://arxiv.org/abs/1212.5701

[39] N. Srivastava, G.E. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (2014), 1929–1958. http://jmlr.org/papers/v15/srivastava14a.html

[40] R. Socher, B. Huval, C.D. Manning, A.Y. Ng, Semantic compositionality through recursive matrix-vector spaces, in Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Association for Computational Linguistics, Jeju Island, 2012, pp. 1201–1211. https://dl.acm.org/citation.cfm?id=2391084

[41] N. Kalchbrenner, E. Grefenstette, P. Blunsom, A convolutional neural network for modelling sentences, in Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Long Papers), Baltimore, 2014, vol. 1, pp. 655–665.

[42] T. Lei, R. Barzilay, T. Jaakkola, Molding CNNS for text: non-linear, non-consecutive convolutions, Indiana Univ. Math. J. 58 (2015), 1151–1186.

[43] R. Zhang, H. Lee, D.R. Radev, Dependency sensitive convolutional neural networks for modeling sentences and documents, in Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, San Diego, 2016, pp. 1512–1521. http://www.aclweb.org/anthology/N16-1177

[44] W. Yin, H. Schütze, Multichannel variable-size convolution for sentence classification, in Proceedings of the Nineteenth Conference on Computational Natural Language Learning, Beijing, 2015, pp. 204–214.