Neural Processing Letters
https://doi.org/10.1007/s11063-019-10017-9

Multi-layer Attention Based CNN for Target-Dependent Sentiment Classification

Suqi Zhang[1] · Xinyun Xu[2] · Yanwei Pang[3] · Jungong Han[4]

Abstract
Target-dependent sentiment classification aims at identifying the sentiment polarities of targets in a given sentence. Previous approaches utilize recurrent neural network with attention mechanism incorporated to model the context and learn key sentiment intermediate representation in relation to a given target. However, such methods are incapable either of modeling complex contexts or of processing data parallelly. To address these problems, we propose, in this paper, a new model that employs a multi-layer convolutional neural network to process the context parallelly and model the context *multiple* times, where the neural network is able to explicitly learn the sentiment intermediate representation via an attention mechanism. Eventually, we integrate these features to form a final sentiment representation, which will be fed into the classifier. Experiments show that our model surpasses the existing approaches on several datasets.

Keywords Target-dependent · Sentiment classification · Multi-layer CNN · Attention mechanism

1 Introduction

With the exponential development of online social and e-commerce platforms such as articles reviews and product reviews, it has become a commonplace for people to express their opinions and comments on the Internet. Most of these opinions and comments contain personal sentiments. Through capturing these sentiments, merchants can know the public's perception of the goods, and make more suitable marketing strategies; consumers can browse the evaluation of the goods and decide whether to purchase or not [1, 2]; the government can grasp the network public's opinion of hot issues and formulate more reasonable policies [3, 4]. Therefore, mining the sentiment information is of great significance.

✉ Suqi Zhang
zhangsuqie@163.com

[1] School of Information Engineering, Tianjin University of Commerce, Tianjin 300134, China

[2] School of Artificial Intelligence, Hebei University of Technology, Tianjin 300401, China

[3] School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China

[4] School of Computing and Communications, Lancaster University, Lancaster LA1 4YW, UK

Published online: 09 March 2019

Roughly speaking, capturing sentiments in texts can be called sentiment analysis and the task has always been a research hotspot. Sentiment analysis is the computational study of people's opinions, attitudes, and emotions toward an entity. The target of sentiment analysis is to find opinions, identify the sentiments they express, and then classify their polarity [5]. Its important subtask is sentiment classification, which is classified into three categories according to positive, negative, and neutral, or divided into five categories according to strong support, support, neutrality, opposition, and strong opposition. In short, depending on the requirements of the task, the division of the categories is also diverse.

There are three main sentiment classification levels: document level, sentence level, and target level [6]. The first two levels' sentiment classifications aim to classify a document and sentence which consider the whole document and sentence as a basic information unit [7–10]. Generally, a sentence contains more than one target. If we consider the sentence as a whole, the polarity judgment is not detailed and clear. Sentiment classification in the target level can be called target-dependent sentiment classification which resolves the polarity classification of a given target. For example, in the sentence "great food but the service was dreadful", the sentiment polarity of the target "food" is positive, while that of "service" is negative. This task has higher practical values and greater challenges so that the research in this area has received more attention in recent years.

Most existing methods utilize RNNs with attention mechanism to learn the context representation and capture the important contextual sentiment features for a given target [11–17]. Among them, the acquisition of target-dependent sentiment information is crucial. For example, in the sentence "great food but the service is dreadful", the sentiment word "dreadful" determines that the sentiment polarity of the target "service" is negative, and the other parts have relatively little impact on the polarity. Therefore, the sentiment information that needs to be captured is "dreadful". In the above work, although RNN can perform well in the sequence task, the input data cannot be processed parallelly because the state of each moment is related to the previous moment. That is, the representation vector of the word "dreadful" contains all the information of the preceding sentence, and this situation should affect the sentiment judgment. As a result, the attention mechanism cannot explicitly obtain the importance of each context word relative to the target, that is, the sentiment information cannot be accurately obtained.

In existing methods, they often generate only one type of context representation by RNN and then treat it as a benchmark to extract useful contextual features. However, the sentiment information representation is diverse, and it is easy to cause the context representation to be not rich enough, and the features capturing from the context representation are relatively simple. For example, in the sentence "This dish is my favorite and I always get it and never get tired of it", the main sentiment features of the target "dish" are the phrases "never get tired of" and "is my favorite". In the sentence "I've had my computer for 2 weeks already and it works perfectly." the main sentiment information of target "works" is the word "perfectly". It can be seen that the expression of sentiment information is diverse and modeling the context once realized through RNN cannot accurately handle different forms of sentiment information. Therefore, it is expected that a variety of context representations can be obtained to realize the extraction of various rich features.

In order to solve the above issues, we propose a model named multi-layer attention based CNN (Mul-AT-CNN). Considering the sentiment polarity of a given target is usually determined by partial context words such as "is my favorite", CNN which can process data parallel and extract the informative n-gram features seems to be a suitable model for our task and also the other natural language processing task [18, 19]. The model regards each convolutional layer's output as a kind of context representation to capture more

context features. The low-level features extracted by CNN are the semantic information of the words and phrases, and the semantic relationships of the distant words and the overall representation of the partial context are obtained as the high-level features. The model introduces the output of each convolution layer into an attention mechanism, which can learn the weight of each context word respectively, and then utilize the weights to calculate a continuous context representation. By integrating the features captured from all kinds of context representation, the final representation can be obtained to predict the sentiment polarity. Considering the context word closer to the target should have more influence on the polarity than a farther one, we apply the location of the context word in the attention mechanism to solve the lack of location information by CNN.

The rest of the paper is organized as follows. Section 2 describes the details of our method. Section 3 provided experimental results as well as the comparison with existing works. Some concluding remarks are drawn in Sect. 4.

2 Related Work

Target-dependent sentiment classification has high practical value and great challenges, and research in this area has received more attention in recent years. And the main methods have transitioned from shallow feature learning to deep semantic learning.

Traditional research methods are based on Syntactic rules, sentiment lexicons and machine learning, such as [20–24]. Hu et al. [21] proposed to use the adjectives as sentiment words to build a sentiment lexicon, and then judge the sentiment polarity of the target according to the polarity of the sentiment words in the lexicon. Kiritchenko et al. [22] used the machine learning method to realize the classification by using the artificially designed features as the basis of the support vector machine (SVM) classifier. Both above traditional methods can effectively leverage text information, but the disadvantage is that it is over d dependent on external resources and artificially designed rules. The approach based on lexicon relies on the accuracy and completeness of the sentiment lexicon, while approach based on machine learning relies on the quality of the selected features. To solve the above problems, the method of using neural networks to learn semantic target-dependent representation and grammatical structures has been given more attention. The difficulty of its method is how to effectively represent the context for a given target.

Dong et al. [11] designed an adaptive recursive neural network (AdaRNN) for target-dependent sentiment classification in twitter, which conveys the sentiments of words towards the target according to the context and syntactic structure. Vo and Zhang [12] automatically obtained rich and effective features using multiple word embeddings, pooling functions, and external sentiment lexicons. Although these methods can capture the sentiment features using neural networks, they still require external information such as syntactic structure and sentiment lexicon. Because RNN seems to be a good fit for sequence tasks, and long short term memory (LSTM) as a variant of RNN can effectively utilize long distance information, Tang et al. [13] regarded the given target as a feature and concatenated it with the context using LSTM. In order to explicitly show the interaction between the target and its context, Zhang et al. [14] proposed bi-directional gated recurrent neural network (Bi-grnn) and the pooling layer to capture the semantics and grammar information of the target and context respectively, then added a three-way gated neural network structure to achieve their interaction.

Since RNN with the attention mechanism firstly proposed in machine translation [25], it has been widely adopted in natural language processing tasks, such as question answering [26]. Late on, its application in sentiment analysis also achieved good results. Wang et al. [15] proposed the LSTM based on attention and target embedding. The attention mechanism allowed the model to pay attention to more important context information of the sentence related to a given target. Tang et al. [16] applied the memory network to this task, using multiple attention mechanisms with external memory to capture the importance of context words for a given target. Ma et al. [17] proposed an interactive attention network (IAN) which combined the target with context using two attention mechanisms to detect important information in the target and context interactively.

3 Methodology

In this section, we describe the proposed approach for target-dependent sentiment classification. Firstly, we give the task definition. After an overview of the approach, we sequentially introduce the multi-layer CNN, attention mechanism and feature integration Finally, we describe how to train the model.

3.1 Task Definition and Notation

Given a sentence $s = \{w_1, w_2, \ldots, w_n\}$ consisting of n words, $\{w_1^\tau, \ldots, w_m^\tau\}$ is a subsequence of s which represents a particular target. The target-dependent sentiment classification aims to obtain the sentiment polarity of the target $\{w_1^\tau, \ldots, w_m^\tau\}$ in s. And the sentiment polarity can be divided into three categories: positive, neutral, negative. For example, in the sentence "great food but the service was dreadful", the sentiment polarity towards target "food" is positive, while the polarity towards target "service" is negative.

Each word in the sentence s can be mapped to a d-dimensional continuous vector called word embedding, for example w_i is mapped to x_i. As mentioned above, the context embeddings can be represented by $\{x_1, x_2, \ldots, x_n\}$ which can also be denoted as a matrix $x_{context} \in \boldsymbol{R}^{d \times n}$. And $x_{target} \in \boldsymbol{R}^d$ is the target embedding which can be obtained by $x_{target} = \frac{x_1^\tau + \cdots + x_m^\tau}{m}$.

In summary, this model uses $x_{context}$ and x_{taregt} as the input to determine the sentiment polarity of the given target.

3.2 Overview of the Approach

In order to classify the sentiment polarity of a given target, we propose a multi-layer convolutional neural network to obtain the semantic and syntactic features of the sentence, and capture the important context information that affects the sentiment polarity through the attention mechanism.

The architecture of the proposed Mul-AT-CNN is shown in Fig. 1.

The model contains multiple compute layers, each consisting of a convolution layer and an attention mechanism. The attention mechanism has two inputs, one is the target embeddings x_{target}, and the other is the context embeddings $x_{context}^l$, where l is the number of layers in CNN. The attention mechanism captures important information in $x_{context}^l$, based on x_{target}.

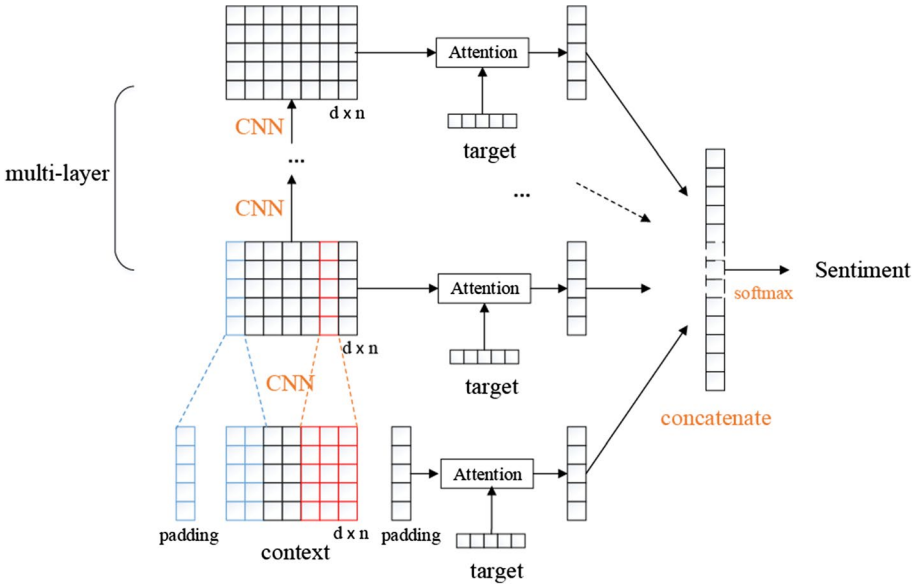


Fig. 1 The architecture of Mul-AT-CNN. In the context matrix, each column represents a d-dimensional word embedding and the number of columns represents the context length n. In the convolution network, vectors of the same color are convoluted separately by d different convolution kernels to form a d-dimensional homochromatic vector. Adding paddings on both sides of the context ensures that the context length remains the same after convolution. The black ellipses and black dashed lines represent the undisplayed multi-layer structures

For the sentence s, the input of the model consists of two parts, one is the context representation $x^0_{context}$ which is regard as $x_{context}$, and the other is the target representation x_{target}. In the first layer, we utilize attention mechanism to capture the important sentiment information of $x^0_{context}$ in relative to x_{target} and then calculate a continuous sentiment representation x^0_{att}. The context representation of the first layer $x^0_{context}$ through the convolution operation can be used as the context representation of the second layer which is regarded as $x^1_{context}$. Then x_{target} and $x^1_{context}$ are input to the attention mechanism to obtain the vector x^1_{att}. In a similar way, we iterate multiple layers to get more abstract context information and more rich sentiment information. The output of each attention layer x^l_{att} is integrated to get a final sentiment representation x_{att}. Finally, inputting x_{att} to softmax layer can obtain the classification result.

3.3 Multi-layer CNN

Compared with RNN, CNN can receive data in parallel, that is, the state at the moment is independent of the previous moment. It can be observed that the sentiment words or phrases affect the sentiment polarity of the sentence in most time. For example, in the sentence "great food but the service was dreadful", "great" determines that the sentiment of the target "food" is positive, while the other parts of the context have little effect on the judgment of polarity. Therefore, using CNN to represent words or phrases is an effective way of context modeling. When we use the attention mechanism to capture important context information, we can explicitly obtain the sentiment vocabulary.

At the same time, a multi-layer CNN can be used to model the context multiple times. Because each kind of context representation expresses different meaning, more profound and abstract contextual information can be obtained from them. Therefore, this paper introduces a multi-layer convolutional neural network to fully exploit the semantic and grammatical features in the sentences and then imports the output of each convolution layer into the attention mechanism to capture important context information for a given target. CNN can generate a holistic representation of the part context covered by the receptive field. When the receptive field is small, a semantic representation of the word and phrase can be realized; as the convolution layer is deepened, the receptive field will also become larger, so that the semantic relationship of the distant words and the overall representation of the partial context can be captured.

The context representation of this layer can be obtained by convolution of the upper layer's representation. To ensure the consistency of context representation dimensions of each layer, the number of convolution kernels is set as the initial word embedding dimension d. Therefore the context representation of the *l-th* layer $x^{l}_{context} \in \boldsymbol{R}^{d\times n}$ can be obtained by convolving the upper layer's context representation $x^{l-1}_{context} \in \boldsymbol{R}^{d\times n}$ with d kernels. Among them, $x^{l}_{context_j} \in \boldsymbol{R}^{1\times n}$ can be obtained by the *j-th* kernel's convolution operation, and the formula of each element in the vector is as:

$$x^{l}_{context_{j_i}} = Relu\left(W^{l-1}_{conv_j} x^{l-1}_{context_{i:i+k-1}} + b^{l-1}_{conv_j}\right), \tag{1}$$

where $x^{l-1}_{context_{i:i+k-1}} \in \boldsymbol{R}^{k\cdot \mathrm{d}}$ is the concatenated vector of $x^{l-1}_{context_i}, \ldots, x^{l-1}_{context_{i+k-1}}$, and k is the kernel size. $W^{l-1}_{conv_j} \in \boldsymbol{R}^{1\times k\cdot \mathrm{d}}$ and $b^{l-1}_{conv_j} \in \boldsymbol{R}^{1\times 1}$ are trainable parameters of the *j-th* convolution kernel.

3.4 Attention Mechanism

The common CNN requires the pooling layer to acquire the main features, such as maxpool which is designed to extract the most important features. However, the features required for this task are contextual information which has a greater impact on the sentiment polarity of a given target. Therefore, we cannot simply acquire the absolute important features, but need to import target to acquire the relative important features. To this end, we introduce the attention mechanism in the model.

In the attention mechanism, the judging whether a feature is important is based on semantic information and location information. The importing of semantic information is to obtain context feature with sentiment polarity for a given target. Considering that context words closer to a target have a greater impact on their sentiment polarity, location information is introduced to focus more on context words that are closer to the target. For example, in the sentence "great food but the service was dreadful", "great" and "dreadful" are both sentimental inclined words. For the target "food", its sentiment polarity is more likely to be positive because "great" is closer to "food".

In this paper, the output of each convolutional layer is imported into the attention mechanism, and attention is used to explore the more important features relative to specific targets. Specifically, the weights of each layer's output vectors are determined, which are used to sum the vectors and calculate a continuous vector. In this way, sentence representations in relation to the specific target's emotional polarity can be obtained.

Based on the target representation x_{target}, learn the weight of each context embedding x_i^l in context representation $x_{context}^l$ and utilize this weight to calculate continuous representation x_{att}^l:

$$x_{att}^l = \sum_{i=1}^{n} \alpha_i^l x_i^l, \tag{2}$$

where α_i^l is the attention parameter based on semantic and location, and its calculation method is:

$$g_i^l = \tanh\left(W_{att}\left[x_i^l; x_{target}\right] + b_{att}\right) \tag{3}$$

$$loc_i = p\left(1 - \frac{l_i}{n}\right) \tag{4}$$

$$\alpha_i^l = \frac{\exp\left(loc_i g_i^l\right)}{\sum_{j=1}^{n} \exp\left(loc_j g_j^l\right)} \tag{5}$$

where g_i^l is the semantic-based attention parameter, and loc_i is the position-based attention parameter. In Eq. (3), $W_{att} \in \boldsymbol{R}^{1\times 2d}$ and $b_{att} \in \boldsymbol{R}^{1\times 1}$ are the weights and offsets of the attention mechanism and need to be trained. It is important to note that the parameters W_{att} and b_{att} of attention are shared in different layers. In Eq. (4), n is the length of the sentence, l_i is the distance from the *i-th* word to the target, and p is the parameter, which needs to be trained.

3.5 Feature Integration

Attention mechanisms capture important features from all kinds of contextual representations respectively. These features are expressions of essential sentiment information in context and can also be used to predict sentiment polarity. However, if we only consider one feature, it is not rich enough for accurate prediction. Therefore, it is necessary to think about how to integrate the various feature representations to make a reasonable contribution to the final prediction. On this basis, we integrate features in our model. The x_{att}^l are concatenated as the vector x_{att}:

$$x_{att} = \left[x_{att}^0; \ldots; x_{att}^{L-1}\right] \tag{6}$$

After that, we introduce x_{att} to a full connected layer and softmax layer for sentiment prediction:

$$p\left(y|x_{context}, x_{target}\right) = softmax\left(W_f x_{att} + b_f\right) \tag{7}$$

where $W_f \in \boldsymbol{R}^{3\times L\cdot \mathrm{d}}$ and $b_f \in \boldsymbol{R}^{3\times 1}$ are learnable parameters.

3.6 Model Training

This model uses an end-to-end backpropagation for training. We need to optimize all the parameters which are from CNN: $[W_{conv_j}^l, b_{conv_j}^l]$, the attention mechanism: $[W_{att}, b_{att}, p]$ and

the softmax layer: $[W_f, b_f]$. The model is optimized by minimizing the cost function with the cross entropy as the cost function.

$$loss = -\sum_{c}\sum_{g}\hat{y}_c^g \log y_c^g + \lambda \|\theta\|_2 \tag{8}$$

where y is the predicted sentiment polarity for a given target, $\hat{y}$ is the actual sentiment polarity, c is the index of the sentence, g is the category index, λ is the L_2 regularization, and θ is the parameter set at the time of regularization.

To avoid overfitting, we use the dropout strategy to randomly drop close half of the neurons in the model. After training, the sentence is tested by importing the context and target into its model and the label with the highest probability represents the expected sentiment polarity of the target.

4 Experiments

The method proposed in this paper is applied to datasets in three different fields for target-dependent sentiment classification, and the validity of the method is verified through experimental results.

4.1 Experimental Setting

We conduct experiments on three benchmark datasets: LAPTOP and REST are from SemEval-2014 [27], containing reviews in laptop and restaurant domain; TWITTER is built by Dong et al. [11], containing twitter. Because these datasets often are be used by the researchers in this task, they have certain authority and the experimental results are also comparable. According to the sentiment polarity, these data can be divided into three categories: positive, negative and neutral. Statistics of the datasets are given in Table 1.

We adopt the accuracy metric in order to evaluate the performance of target-dependent sentiment classification, which is defined as $Acc = \frac{T}{N}$, where T is the number of correctly predicted samples, N is the total number of samples. Generally, a well-performed system has a higher accuracy.

We use Glove vectors [28] with 300 dimensions as pre-trained word embeddings. For unregistered words, use the uniform distribution U (−0.25, 0.25) to initialize. The word vector dimension and the target representing dimension are 300. The convolution windows' size is 2. Appropriate padding is applied to the input of each convolution layer to make the output length consistent with the input length. In order to ensure the weight sharing

Table 1 Experimental dataset statistic

Dataset	Positive	Negative	Neutral
LAPTOP-train	987	866	460
LAPTOP-test	341	128	169
REST-train	2164	805	633
REST-test	728	196	196
TWITTER-train	1567	1563	3127
TWITTER-test	174	174	346

Table 2 Hyper-parameter values

Parameter	Parameter description	Value
d	Dimension of vector	300
k	Convolution kernel size	2
Drop	Dropout rate	0.5
Batch	Batch size	128
lr	Initial learning rate	0.01

Table 3 Experimental results

Model	LAPTOP (%)	REST (%)	TWITTER (%)
LSTM	66.45	74.28	–
TD-LSTM	68.13	75.63	66.73
ATAE-LSTM	68.70	77.20	–
MemNet	72.26	77.16	70.23
CNN-MemNet	73.83	78.77	70.89
IAN	72.10	78.60	–
Mul-AT-CNN	75.39	79.46	71.25

of the attention layer, vector dimension of the hidden layer is consistent with the dimension of the word vector. That is, the number of the convolution kernels in a layer is 300. All weight matrices are initialized with the uniform distribution U (−0.01, 0.01) and the biases are initialized as zeros. The batch size is 128 and the initial learning rate is 0.01. To reduce overfitting, dropout is set to 0.5. Statistics of the hyper-parameter values are given in Table 2.

4.2 Comparison with Baseline Methods

We compare our model with several existing baselines, including LSTM, TD-LSTM, ATAE-LSTM, MemNet, CNN-MemNet, IAN.

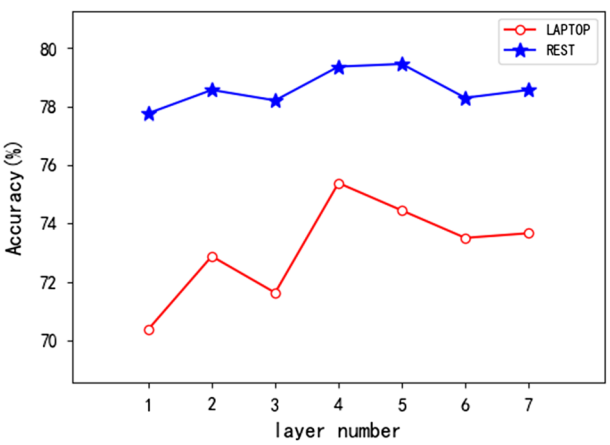


Fig. 2 Effects of layers number. The horizontal axis represents the number of network layers, and the vertical axis represents the accuracy. In the table, the red line represents the experimental results in LAPTOP, and the blue line represents the experimental results in REST. (Color figure online)

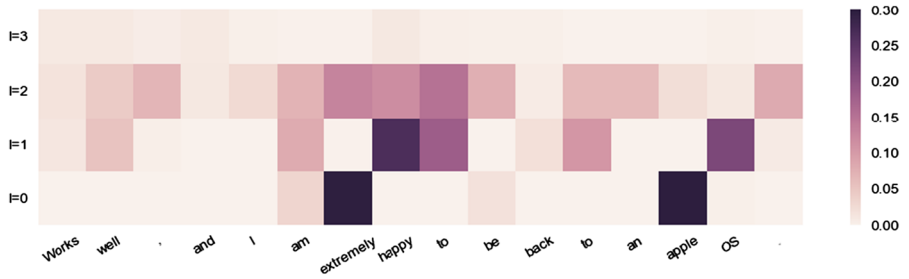
Fig. 3 Attention visualizations. The color depth expresses the importance degree of the weight in attention vector α. The row refers to the attention weight of a layer. The weights of the first layer are placed at the bottom, and the weights of the other layers are added upwards in turn. (Color figure online)

- *LSTM* It only uses one LSTM network to model the context and get the hidden state of each word. After that, the average value of all the hidden states is regarded as final representation and fed to a softmax function to estimate the probability of each sentiment label.
- *TD-LSTM* [13] It employs a forward LSTM and backward LSTM to represent the left context with the target and the right context with the target respectively, and then predicts the sentiment polarity according to the concatenated last hidden states.
- *ATAE-LSTM* [15] It introduces the attention mechanism and target embedding to LSTM. It connects each word embedding with the target embedding to strengthen the effects of targets in hidden states, then uses attention mechanism to pay more attention to the important embeddings regarded to the target from the hidden states.
- *MemNet* [16] It employs the memory network for this task which regards the context as the external memory and uses the multiple attention mechanisms to capture the sentiment feature from this memory repeatedly.
- *CNN-MemNet* Based on the MemNet model, the external memory was replaced by a multi-layer CNN. The number of calculated layers corresponds to the number of convolution layers. Attention mechanism can capture the sentiment feature from the CNN memory.
- *IAN* [17] It employs two LSTMs to model the context and target. The attention mechanisms interactively learn key feature representations in the context and target, and then concatenate these representations for prediction.

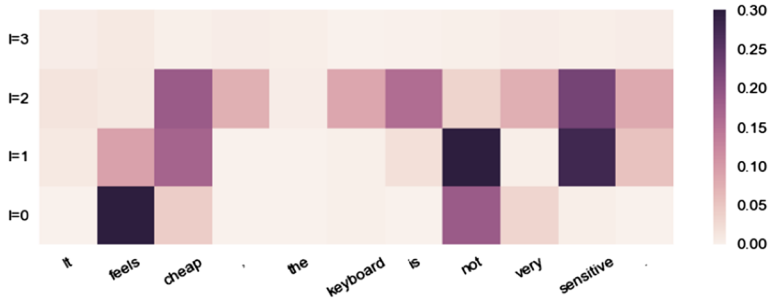
Table 3 shows the experimental results comparison Mul-AT-CNN with baselines. We run the released codes of TD-LSTM to generate results, since the paper only reported result on TWITTER. We also rerun MemNet on our datasets and evaluate it with accuracy.

It can be seen that the performance of LSTM is the worse because it doesn't consider the target and treats it equally with other context words. Thus, LSTM can only acquire the sentiment polarity of the entire sentence. And TD-LSTM performing better than LSTM also indicates that the target is the key information for our task. ATAE-LSTM performs better than TD-LSTM, mainly because the attention mechanism can capture the sentiment information corresponding to the target and achieve the information interaction from the target to the context. MemNet doesn't utilize the RNN to model context representation, which directly regards the original context word embeddings as context representation. The performance which is better than ATAE-LSTM means that the task can be completed well without using RNN. CNN-MemNet replaces the context representation in MemNet with a multi-layer CNN, which implements a variety of modeling representations. Its good performance shows that it is effective to use CNN to model context representations. Further, IAN performance well shows the more consideration should be given to model target.

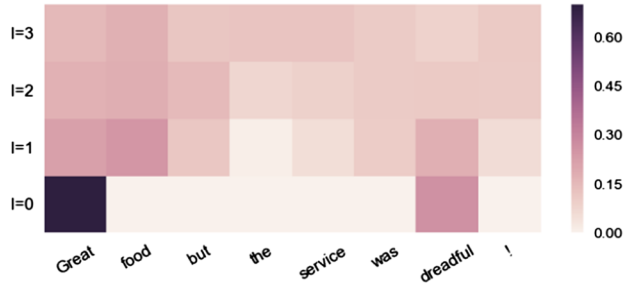
Our approach outperforms the baselines on these datasets, especially achieving a 3% improvement in LAPTOP, but the growth in TWITTER is the slowest because of its tweets with more ungrammatical sentences. There may be a long distance between the target and the sentiment word in context so that the model cannot fit the data well. MemNet and CNN-MemNet both capture sentiment information in context representations with multi-layer attention mechanism. However, these attention mechanisms capture effective



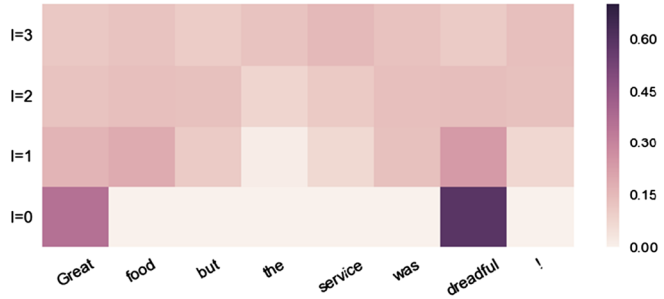
(a) The target of this sentence is apple OS and the sentiment polarity is positive



(b) The target of this sentence is keyboard and the sentiment polarity is negative



(c) The target of this sentence is food and the sentiment polarity is positive



(d) The target of this sentence is service and the sentiment polarity is negative

sentiment feature not only in regard to target but also based on the acquired sentiment information, which may introduce more irrelevant information. The fact that our method performs better than them shows that capturing sentiment information directly based on the target is a better choice with the stronger interpretability. And our model can receive input data in parallel, so we can obtain important context information without influence from the previous context. And the model realizes multiple modeling of the context, so it can obtain more abstracted grammatical and semantic information.

4.3 Effects of Layers Number

As Mul-AT-CNN involves multiple layers, we explore the effect of the layer number L. We conduct experiments with 1–7 layers in LAPTOP and REST respectively. The experimental results are shown in Fig. 2. We can observe that the best performances are achieved when the model contains four and five layers respectively. It shows that multiple layers generally lead to better performance, especially when L is less than five. In contrast, when L is small such as 1 or 2, the performance is poor, which shows that the context representation is not sufficient to include the important sentiment feature. It reveals that more network layers do not necessarily get the performance better. For example, the 7-layer network's performance is not as good as 4-layers' on the LAPTOP. This is because as the number of layers increases, the model parameters increase which make model difficult to train and less generalizable.

4.4 Attention Visualization

We select some samples from LAPTOP. In order to observe the context information captured by the attention mechanism, we obtain the attention weights in Eq. (5) and visualize the attention weights accordingly. Figure 3 shows some sample cases. The color depth indicates the importance degree of the weight in the attention vector, the darker the more important. Each row in Fig. 3 represents the weight distribution of one layer. The weights of the first layer are placed at the bottom of the figure, and the weights of the other layers are added upwards in turn. Because we use a 4-layer network to evaluate performance, there are 4 rows in the figure.

The sample in Fig. 3a is "Works well, and I am extremely happy to be back to an apple OS" whose target is "apple OS". From the figure, it can be seen that the target word "apple" and the degree modifier "extremely" are mainly given the high weights in the first layer. In the second layer, it finds sentiment information such as "work well" and "extremely happy" and complete target "apple OS". The key information found by the third and fourth layers is a representation of the partial context centered on the captured sentiment vocabulary. Finally, sentiment polarity of positive can be judged by the information available above, such as "work well" and "extremely happy".

The sample in Fig. 3b is "It feels cheap, the keyboard is not very sensitive" whose target is "keyboard". In the first layer the negative word "not" and the sentiment information "feels" are paid more attention. In the second layer, sentiment phrase such as "feels cheap" and "sensitive" are found. Same as the previous example, the key information found by the third and fourth layers is a representation of the partial context such as "the keyboard is", "not very sensitive". So the sentiment polarity is negative that can be determined by the important information available above, such as "feels cheap", "not very sensitive".

Unlike the previous two samples, the following samples show the weight distribution when there are multiple targets in the same sentence. The sentence is "great food but the service was dreadful", where the target in Fig. 3c is "food" and the target in Fig. 3d is "service". From the Fig. 3c, it can be seen that the first layer of the model focuses on the sentiment words "great" and "dreadful", but pays more attention to "great". And in the latter layer, the model pays more attention to "great" with its surrounding words than "dreadful". Conversely, in Fig. 3d, the model pays more attention to "dreadful" with its surrounding words than "great". Therefore, it can be seen that the model can effectively distinguish the sentiment information based on different targets.

In summary, attention can effectively obtain target information and sentiment information in the first few layers. And the information found in high layers is representation of the partial context centered on the captured vocabularies in low layers. Therefore, the final representation includes not only a few sentiment words, but an overall representation of the partial context, even the sentence which contains enough rich information to solve the task.

5 Conclusion

In this paper, we have proposed a multi-layer attention-based CNN, which uses multi-layer CNN to model a variety of context representations and explores the sentiment information in context representation through attention mechanisms repeatedly. Compared with the RNN-based method, our method can obtain sufficient context information through multiple modeling of the convolutional layer and can explicitly capture important sentiment vocabulary because of inputting data parallelly. Experimental results show that our model achieves an improved performance on a few benchmarks. However, this model does not consider the complicated modeling of the target, which will be one of the future works. Another possible future work will be applying our multi-layer attention model to other application domains, such as face recognition [29–31], object tracking [32–35], image saliency detection [36–38] and video retrieval [39–41].

Acknowledgements This research was supported by the National Natural Science Foundation of China Grant 61802282, the National Natural Science Foundation of Hebei Province through the Key Program under Grant F2016202144, the Science and Technology Program of Hebei Province Grant 17210305D.

References

1. Zhao S, Ding G, Gao Y, Han J (2017) Approximating discrete probability distribution of image emotions by multi-modal features fusion. In: Proceeding of the twenty-sixth international joint conference on artificial intelligence, pp 4669–4675
2. Zhao S, Yao H, Gao Y, Ji R, Ding G (2016) Continuous probability distribution prediction of image emotions via multitask shared sparse regression. IEEE Trans Multimed 19(3):632–645
3. Zhao S, Yao H, Gao Y, Ding G, Chua T (2018) Predicting personalized image emotion perceptions in social networks. IEEE Trans Affect Comput 9(4):526–540
4. Zhao S, Gao Y, Ding G, Chua T (2018) Real-time multimedia social event detection in microblog. IEEE Trans Cybern 48(11):3218–3231
5. Medhat W, Hassan A, Korashy H (2014) Sentiment analysis algorithms and applications: a survey. Ain Shams Eng J 5(4):1093–1113
6. Liu B (2012) Sentiment analysis and opinion mining. Synth Lect Hum Lang Technol 5(1):1–167
7. Glorot X, Bordes A, Bengio Y (2011) Domain adaptation for large-scale sentiment classification: a deep learning approach. In: Proceedings of the twenty-eighth international conference on machine learning, pp 513–520

8. Moraes R, Valiati JF, Neto WPG (2013) Document-level sentiment classification: an empirical comparison between SVM and ANN. Expert Syst Appl 40(2):621–633
9. Santos CND, Gattit M (2014) Deep convolutional neural networks for sentiment analysis of short texts. In: Proceedings of the twenty-fifth international conference on computational linguistics: technical papers, pp 69–78
10. Yu J, Jiang J (2016) Learning sentence embeddings with auxiliary tasks for cross-domain sentiment classification. In: Proceedings of the 2016 conference on empirical methods in natural language processing, pp 236–246
11. Dong L, Wei F, Tan C, Tang D, Zhou M, Xu K (2014) Adaptive recursive neural network for target-dependent twitter sentiment classification. In: Proceedings of the fifty-second annual meeting of the association for computational linguistics (volume 2: short papers), vol 2, pp 49–54
12. Vo DT, Zhang Y (2015) Target-dependent twitter sentiment classification with rich automatic features. In: Proceedings of the twenty-forth international joint conference on artificial intelligence, pp 1347–1353
13. Tang D, Qin B, Feng X, Liu T (2016) Effective LSTMs for target-dependent sentiment classification. In: Proceedings of the international conference on computational linguistics: technical papers, pp 3298–3307
14. Zhang M, Zhang Y, Vo DT (2016) Gated neural networks for targeted sentiment analysis. In: Proceedings of AAAI conference on artificial intelligence, pp 3087–3093
15. Wang Y, Huang M, Zhao L et al (2016) Attention-based LSTM for aspect-level sentiment classification. In: Proceedings of the 2016 conference on empirical methods in natural language processing, pp 606–615
16. Tang D, Qin B, Liu T (2016) Aspect level sentiment classification with deep memory network. In: Proceedings of the 2016 conference on empirical methods in natural language processing, pp 214–224
17. Ma D, Li S, Zhang X, Wang H (2017) Interactive attention networks for aspect-level sentiment classification. In: Proceeding of the twenty-sixth international joint conference on artificial intelligence, pp 4068–4074
18. Gehring J, Auli M, Grangier D, Yarats D, Dauphin YN (2017) Convolutional sequence to sequence learning. arXiv preprint arXiv:170503122
19. Kim Y (2014) Convolutional neural networks for sentence classification. In: Proceedings of the 2014 conference on empirical methods in natural language processing, pp 1746–1751
20. Ding X, Liu B (2007) The utility of linguistic rules in opinion mining. In: Proceedings of the thirtieth annual international ACM SIGIR conference on research and development in information retrieval. ACM, pp 811–812
21. Hu M, Liu B (2004) Mining and summarizing customer reviews. In: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp 168–177
22. Kiritchenko S, Zhu X, Cherry C, Mohammad S (2014) NRC-canada-2014: detecting aspects and sentiment in customer reviews. In: Proceedings of the eighth international workshop on semantic evaluation (SemEval 2014), pp 437–442
23. Nasukawa T, Yi J (2003) Sentiment analysis: capturing favorability using natural language processing. In: Proceedings of the second international conference on knowledge capture. ACM, pp 70–77
24. Jiang L, Yu M, Zhou M, Liu X, Zhao T (2011) Target-dependent twitter sentiment classification. In: Proceedings of the forty-ninth annual meeting of the Association for Computational Linguistics: human language technologies-volume 1. Association for Computational Linguistics, pp 151–160
25. Bahdanau D, Cho K, Bengio Y (2014) Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:14090473
26. Wang W, Yang N, Wei F, Chang B, Zhou M (2017) Gated self-matching networks for reading comprehension and question answering. In: Proceedings of the fifty-fifth annual meeting of the association for computational linguistics (volume 1: long papers), vol 1, pp 189–198
27. Pontiki M, Galanis D, Pavlopoulos J, Papageorgiou H, Androutsopoulos I, Manandhar S (2014) SemEval-2014 task 4: aspect based sentiment analysis. In: Proceedings of international workshop on semantic evaluation, pp 27–35
28. Pennington J, Socher R, Manning C (2014) Glove: global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing, pp 1532–1543
29. Wang N, Gao X, Sun L, Li J (2017) Bayesian face sketch synthesis. IEEE Trans Image Process 26(3):1264–1274
30. Wang N, Gao X, Li J (2018) Random sampling for fast face sketch synthesis. Pattern Recogn 76:215–227

31. Wang N, Gao X, Sun L, Li J (2018) Anchored neighbourhood index for face sketch synthesis. IEEE Trans Circuits Syst Video Technol 28(9):2154–2163
32. Lan X, Zhang S, Yuan P, Chellappa R (2018) Learning common and feature-specific patterns: a novel multiple sparse representation based tracker. IEEE Trans Image Process 27(4):2022–2037
33. Han J, Pauwels P, de Zeeuw P, de With P (2012) Employing a RGB-D sensor for real-time tracking of humans across multiple re-entries in a smart environment. IEEE Trans Consum Electron 58(2):255–263
34. Lan X, Ma J, Yuan P, Chellappa R (2015) Joint sparse representation and robust feature-level fusion for multi-cue visual tracking. IEEE Trans Image Process 24(12):5826–5841
35. Ding G, Chen W, Zhao S, Han J, Liu Q (2018) Real-time scalable visual tracking via quadrangle kernelized correlation filters. IEEE Trans Intell Transp Syst 19(1):140–150
36. Yan C, Xie H, Chen J, Zha Z, Hao X, Zhang Y, Dai Q (2018) A fast Uyghur text detection for complex background images. IEEE Trans Multimed 20(12):3389–3398
37. Zhang D, Meng D, Han J (2017) Co-saliency detection via a self-paced multiple-instance learning framework. IEEE Trans Pattern Anal Mach Intell 39(5):865–878
38. Cheng G, Zhou P, Han J (2016) Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images. IEEE Trans Geosci Remote Sens 54(12):7405–7415
39. Yan C, Xie H, Yang D, Yin J, Zhang Y, Dai Q (2018) Supervised hash coding with deep neural network for environment perception of intelligent vehicles. IEEE Trans Intell Transp Syst 19(1):284–295
40. Wu G, Han J, Guo Y, Liu L, Ding G, Ni Q, Shao L (2019) Unsupervised deep video hashing via balanced code for large-scale video retrieval. IEEE Trans Image Process 28(4):1993–2007
41. Wu G, Han J, Lin Z, Ding G, Zhang B, Ni Q (2019) Joint image-text hashing for fast large-scale cross-media retrieval using self-supervised deep learning. IEEE Trans Ind Electron. https://doi.org/10.1109/tie.2018.2873547 **(in press)**