Contextualized Word Representations with Effective Attention for Aspect-based Sentiment Analysis

Zixuan Cao[1], Yongmei Zhou[1,2(✉)], Aimin Yang[1,3], and Jiahui Fu[3]

[1] School of Information Science and Technology, School of Cyber Security, Guangdong University of Foreign Studies, Guangzhou, China
[2] Eastern Language Processing Center, Guangdong University of Foreign Studies, Guangzhou, China
[3] School of Business, Guangdong University of Foreign Studies, Guangzhou, China
{niketim,yongmeizhou,chaihui_fu}@163.com, amyang18@qq.com

Abstract. Aspect-based sentiment analysis (ABSA) aims at identifying sentiment polarities towards aspect in a sentence. Attention mechanism has played an important role in previous state-of-the-art neural models. However, existing attention mechanisms proposed for aspect based sentiment classification mostly focus on identifying the sentiment words, without considering the relevance of such words with respect to the given aspects in the sentence. To solve this problem, we propose a new architecture, self-attention with co-attention (SACA) for aspect-based sentiment analysis. Self-attention is capable of conducting direct connections between arbitrary two words in context from a global perspective, while co-attention can capture the word-level interaction between aspect and context. Moreover, previous works simply averaged aspect vector to learn the attention weights on the context words, which may bring information loss if the aspect has multiple words. To address the problem, we employ the pre-trained contextual word embeddings and character-level word embeddings as word representation. We evaluate the proposed approach on three datasets, experimental results demonstrate that our model outperforms the state-of-the-art on all three datasets.

Keywords: Aspect-based sentiment analysis · Self attention · Co-attention.

1 Introduction

Aspect-based sentiment classification is an important task in fine-grained sentiment analysis. The goal of ABSA is to predict the sentiment polarity of the sentence for the given aspect. For example, in the sentence "*The **food** is usually good but it certainly isn't a relaxing **place** to go.*", the user mentions two targets "***food***" and "***place***", and expresses positive sentiment over the target "*food*", but negative sentiment over "*place*". Compared to sentence or document level sentiment analysis, the challenge of aspect level sentiment analysis is to differentiate the emotions of different targets.

Traditional methods [1,2] for ABSA mainly focus on feature engineering (such as bag-of-words and sentiment lexicons) to train a classifier for sentiment classification. However, traditional methods are mostly based on manual work, which requires a lot of time and manpower. Therefore, many neural network-based models have been proposed in recent years. Recurrent Neural Networks (RNNs) with attention mechanism, firstly proposed in machine translation [3], has been successfully used in many NLP tasks. In the task of ABSA, many works [4,5,6,7] employ attention mechanism to measure the semantic relatedness between context word and the target. However, many approaches [6,8] simply average aspect vectors to learn the attention weights on the context words. This may work fine for targets that only contain one word but may fail to capture the semantics of more complex expressions. For example, we cannot obtain the representation for "*hot pot*" by averaging the word vectors of "*hot*" and "*pot*". "*hot*" would be close to words like *"temperature"* and *"pot"* would be close to words like *"cooking tools"*. The averaged word vector could be distant from the actual vector for "*hot pot*".

To address this problem, we first use the contextualized word embedding named ELMo [9]. The traditional word embeddings, such as Word2vec [14], Glove [15], only have one representation per word, and therefore cannot capture how the meaning of each word can change based on surrounding context. However, ELMo analyses words within the context that they are used and also allowing the model to form representations of out-of-vocabulary (OOV) words. On this basis we further employ a co-attention mechanism to characterize the word-level interactions between aspect and context words.

Though bidirectional recurrent neural networks can model long distance dependencies, it cannot conduct direct connections between arbitrary two words. To better exploit the global dependencies of the sequential input (i.e., context and aspect), we employ the self-attention mechanism, which has been introduced to machine translation by Vaswani [10], and it is very expressive and flexible for modeling long-range and local dependencies.

In this paper, we introduce a novel neural network named **S**elf-**A**ttention with **C**o-**A**ttention (SACA) for aspect-term sentiment analysis. It consists of a self-attention blocks to better exploit the global and local dependencies, and an aspect-context co-attention block to attend target and textual information for improving sentence representation. To evaluate the proposed approach, we conduct experiments on three datasets: SemEval 2014 dataset, containing reviews of restaurant domain and laptop domain, the third one is a tweet collection. The experimental results demonstrate the effectiveness of our proposed model.

2 Related Work

Most of the early methods adopted supervised learning methods with extensive manually designed features such as sentiment lexicon, n-grams, and dependency information, then training a sentiment classifier [1,2,11]. Kiritchenko et al. [1] proposed to use SVM based on n-gram features. Vo and Zhang [11] used pool-

ing functions to extract features from sentiment-specific word embeddings and sentiment lexicons. However, these methods are labor-intensive and they failed to model the semantic relatedness between a target and its context information.

With the advances of deep learning methods, various neural models [6,12,13] have been proposed for automatically encoding sentence features as continuous and low-dimensional vectors without feature engineering. Tang et al. [12] proposed the target-dependent LSTM (TD-LSTM) and target connection LSTM (TC-LSTM) to model the interaction between target and the whole sentence. The RNN based models have achieved promising results, but they do not take into account the relatedness between the context words and the given aspect. To solve this problem, attention based neural methods [5,8,13] have been successfully applied to the ABSA problem due to their ability to explicitly capture the importance of context words. Tang et al. [8] computed the sentence representation by stacking multiple layer of attention. Ma et al. [5] proposed a bidirectional attention model IAN to interactively learning attention in the context and target, and generate the representation for target and context separately. More recently, Fan et al. [13] used the fine-grained and coarse-grained attention mechanisms to capture the interaction between aspect and context.

3 Model

3.1 Task Definition

Given a sentence $s = (w_1, w_2, ..., w_n)$ consisting of n words, and an aspect occurring in the sentence $q = (a_1, a_2, ..., a_m)$ consisting of a subsequence of m continuous words from s. Aspect-based sentiment classification aims to determine sentiment polarity of the sentence s towards each aspect q.

We present the overall architecture of the proposed **S**elf-**A**ttention and **C**o-**A**ttention (SACA) in Fig.1. It consists of the word representation layer, the contextual layer, the attention layer and output layer.

3.2 Word Representation Layer

Our model encodes words of an input sentence or aspect as a combination of three different types of embeddings, which can be listed as follows:
Pre-trained word embeddings: We use the pre-trained Glove embedding [15] for initialization and keep fixed during the training process.
Pre-trained contextual word embeddings: We use pre-trained ELMo [9] embeddings. These representations are extracted from the hidden states of a bidirectional language model. ELMo embeddings have been shown to give state-of-the-art results in many NLP tasks.
Character-level word embeddings: We use character-level word embeddings to extract character-level features, which have been shown to be helpful to deal with out-of-vocab (OOV) tokens.

Formally, given a sentence s, we suppose the Glove, ELMo and character-level word representations of the sentence are $W \in R^{n*d_w}$, $E \in R^{n*d_e}$ and

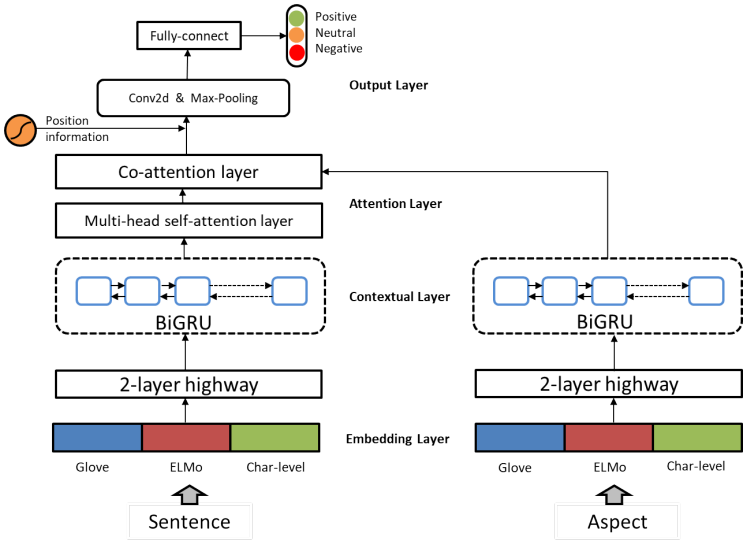

Fig. 1. The architecture of the proposed model.

$C \in R^{n*r*d_c}$, respectively. Where n denotes the sentence length, r denotes the word length, d_w, d_e and d_c represent the Glove embedding dimension, ELMo embedding dimension and character-level embedding dimension, respectively. Specifically, we firstly use a convolutional neural network (CNN) to encode the character C, and then concatenate the output of convolutional feature with Glove embedding and ELMo embedding to form the final word representations for the review sentence $e_s \in R^{n*|e|}$ and aspect $e_q \in R^{m*|e|}$, where n,m are the length of the review sentence and aspect, $|e|$ is the final embedding size (including all three components).

$$e = Concat(W, E, Conv2D(C)) \tag{1}$$

After the concatenation of three embedding component, we further feed word embeddings into a two-layer highway network [21] with tanh output activation.

$$\tilde{e} = \tanh(2 \sim highway(e)) \tag{2}$$

3.3 Contextual Layer

We employ a bidirectional recurrent neural network (RNN) as encoder to sequentially process each word in s and q, we chose to use Gated Recurrent Unit (GRU) [16] in our experiment since it performs similarity to LSTM but is computationally cheaper.

$$h_{s_i} = BiGRU(h_{s_{i+1}}, h_{s_{i-1}}, \tilde{e}_{s_i}) \qquad h_{q_j} = BiGRU(h_{q_{j+1}}, h_{q_{j-1}}, \tilde{e}_{q_j}) \tag{3}$$

We employ the BiGRU separately and get the context hidden output $H_s = (h_{s_1}, h_{s_2}, ..., h_{s_n})$ and the aspect hidden output $H_q = (h_{q_1}, h_{q_2}, ..., h_{q_m})$

3.4 Attention Layer

The attention layer includes three sub-layers: (1) self-attention layer; (2) co-attention layer; (3) position-aware attention layer.

Self-Attention Layer We adopt the multi-head self-attention mechanism [10] to capture long range dependencies and inner structure of the sequential input. We regard Q, K and V as query matrix key matrix and value matrix, respectively. The scaled dot product attention can be described as follows:

$$Att(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{4}$$

Where n is the length of the sequence, and d_k denotes feature dimension.

In order to attend the information from different representation subspaces, multi-head self-attention concatenates the output of all scale l dot product attention models and then project concatenated feature to a fix dimensional feature.

$$head_i = Att(H_s W_i^Q, H_s W_i^K, H_s W_i^V) \tag{5}$$

$$\tilde{H} = (head_1 \oplus ... \oplus head_l)W_o \tag{6}$$

Where $H_s = (h_{s_1}, h_{s_2}, ..., h_{s_n})$ denotes the context output of BiGRU, n is the length of context sequence, W_i and W_o are trainable projection parameters. In order to avoid the loss of information in multi-head self-attention operations, we add a residual connection to $\tilde{H}$ and then apply a layer normalization:

$$O = LayerNorm(\tilde{H} + H_s) \tag{7}$$

Co-Attention Layer Considering the example "*Straight-forward, no surprises, very decent Japanese food*", the word "*food*" in aspect "*Japanese food*" should have more effect on the context words compared with word "*Japanese*". Accordingly, the context words should pay more attention on "*food*" instead of "*Japanese*". To solve this problem, we employ co-attention mechanism that attends to the review sentence and aspect simultaneously.

Formally, we first construct an alignment matrix:
$L_{ij} = W_l([O; H_q; O - H_q; O * H_q])$, where L_{ij} denotes the similarity between i-th context word and j-th aspect word, O is the output of the self-attention layer, H_q is the aspect hidden output, [;] denotes the concatenation operator and * denotes the element-wise multiplication. The alignment matrix is normalized row-wise to produce the context-to-aspect attention weight A_q, and normalized column-wise to produce the aspect-to-context attention A_s across the aspect for each word in the context:

$$A_q = softmax(L) \qquad A_s = softmax(L^T) \tag{8}$$

And then we apply the computed attention map to the A_q to the aspect feature H_q to obtain the attended context feature $C_s = A_q H_q$, we also compute the attented aspect feature $C_q = A_s H_s$. Finally, we define co-attention representation of the aspect and context: $C = [C_s; (A_s^T C_q)]$.

In order to capture all of the information and highlight the significant features, we use a Bi-GRU to get the fusion of temporal information:

$$u_t = BiGRU(u_{t-1}, u_{t+1}, C_i) \tag{9}$$

We define $U = [u_1, ..., u_n]$, which is the aspect-aware sentence representation.

Position-aware Attention Layer Following an important observation found in [6,12] that sentiment words towards the aspect is more likely to be expressed near the aspect. For example, "*service*" in "*The price is reasonable although the service is poor.*" may be associated with opinion word "*poor*". Specifically, we calculate the target position weight between the i-th context and the aspect term as follows:

$$p_i = \begin{cases} 1 - \frac{m_0 - i}{n}, & i < m_0 \\ 0, & m_0 \leq i \leq m_0 + m \\ 1 - \frac{i - (m_0 + m)}{n}, & i > m_0 + m \end{cases} \tag{10}$$

Where m_0 is the index of the first aspect word, n and m are the length of sentence and aspect, respectively. We use the position weight to attend the output of the previous layer: $\hat{u_i} = u_i * p_i$. Obviously, the i-th context word closer to a aspect term with a large position weight. To capture the most informative features, we feed the weight $\hat{u}$ to the convolutional layer with max pooling to generate the feature map:

$$c_i = ReLU(w_{conv}^T \hat{u}_{i:i+s-1} + b_{conv}) \tag{11}$$

$$v_{max} = MaxPooling(c_1, c_2, ..., c_n) \tag{12}$$

Where s is the kernel size, w_{conv} and b_{conv} are trainable parameters of convolution layer.

3.5 Output Layer

Finally, we fed v_{max} to a $softmax$ layer for determining the aspect sentiment polarity:

$$p = softmax(W_p v_{max} + b_p) \tag{13}$$

Where p is the probability distribution for the polarity of aspect sentiment, W_p and b_p are learnable parameters.

Loss Function To train our model, we use traditional categorical cross entropy loss with L_2-regularizer as loss function:

$$loss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{i,j}\log p_j + \lambda\|\theta\|^2 \tag{14}$$

Where N is the number of samples, C is the number of sentiment categories, λ is the regularization weight, θ is the set of trainable parameters, $y_{i,j}$ is a one hot class vector for the j-th class and p_j is the predicted probability for the j-th class.

4 Experiments

We conduct experiments on three benchmark datasets, as shown in Table 1. The first two are from SemEval 2014 Task 4 [17], containing customer reviews on restaurants and laptop. The third one is a collection of tweets, built by Dong [18]. Each review has one or more aspect with their corresponding polarities. The polarity of an aspect can be ***Positive***, ***Netural*** and ***Negative***. Evaluation metrics are Accuracy and Macro-F1, which is widely used in previous works. The main hyper-parameters of our model are listed in Table 2. Adam [19] is adopted as the optimizer with an initial learning rate of 0.0005. In order to prevent overfitting, we apply dropout of 0.3 to all the representation layers. The model is build on Pytorch platform.

Table 1. The statistics of the three datasets

Dataset	#Positive		#Netural		#Negative	
	Train	Test	Train	Test	Train	Test
Restaurant	2164	728	637	196	807	196
Laptop	994	341	464	169	870	128
Twitter	1561	173	3127	346	1560	173

Table 2. Hyper-parameter settings

Symbol	Descriptions	Size
n	sentence max length	64
m	aspect max length	5
d_w	glove word embedding size	300
d_e	ELMo embedding size	1024
d_c	character embedding size	16
d_h	Bi-GRU hidden size	400
l	heads of self-attention	8

4.1 Model Comparisons

We compare our model against the following baseline methods:
(1) **Feature + SVM** [1]: The classic SVM model using a series of manual features.
(2) **LSTM** [4]: An LSTM network is built on top of word embeddings, the mean of all the hidden outputs from LSTM is taken as the sentence representation.
(3) **TD-LSTM** [12]: It uses a forward LSTM and a backward LSTM to model context before and after the aspect.
(4) **AE-LSTM** and **ATAE-LSTM** [4]: AE-LSTM is a simple LSTM model incorporating the target embedding as input. ATAE-LSTM is developed based on AE-LSTM and uses attention mechanism to generate the final representation from hidden states.
(5) **MemNet** [8]: It applies multi-hop attention on the memory stacked by input word embeddings and predicts sentiment based on the top context representation.
(6) **IAN** [5]: It adopts two LSTMs with attention mechanism to generate representations of aspect and context separately by interactive learning.
(7) **IARM** [20]: It adopts a GRU and attention mechanism to generate the aspect-aware sentence representations, and also incorporate the neighboring aspects related information into the sentiment classification of the target aspect using memory networks.
(8) **MGAN** [13]: It adopts coarse-grained and fine-grained attention mechanism for sentence representation.

Table 3. The performance comparisons of different methods on three datasets, where the results of baseline methods are retrieved from the original papers. "-" means this result is not available. The best performances are marked in bold.

Methods	Laptop		Restaurant		Twitter	
	Acc.	Macro-F1	Acc.	Macro-F1	Acc.	Macro-F1
Feature+SVM	0.7049	-	0.8016	-	0.6340	0.6330
LSTM	0.6679	0.6402	0.7523	0.6421	-	-
TD-LSTM	0.6825	0.6596	0.7537	0.6451	0.6662	0.6401
AE-LSTM	0.6890	-	0.7660	-	-	-
ATAE-LSTM	0.6870	-	0.7720	-	-	-
MemNet	0.7033	0.6409	0.7816	0.6583	0.6850	0.6691
IAN	0.7210	-	0.7860	-	-	-
IARM	0.7380	-	0.8000	-	-	-
MGAN	0.7539	0.7247	0.8125	0.7194	0.7254	0.7081
SACA	**0.7633**	**0.7292**	**0.8205**	**0.7310**	**0.7268**	**0.7103**

4.2 Main Results

From Table 3, we can have the following observations: (1) Compared with all other neural baselines, our model achieves significant improvement on both accuracy and macro-F1 scores for three datasets. (2) Feature-based SVM is still a strong baseline, which demonstrates the importance of high quality features for aspect based sentiment analysis. Our approach can achieve competitive results without relying on so many manually-designed features. (3) The attention based models (ATAE-LSTM, MemNet, IAN, IARM, MGAN and SACA) perform better than non-attention based models (LSTM, TD-LSTM and AT-LSTM), one main reason maybe the attention mechanism can make the model notices important parts of a sentence for a given aspect. (4) IARM achieves slightly better results with the previous RNN-based methods, which employ memory network to model the dependency of the target aspect with other aspects in the sentence. (5) MGAN achieves the best performances among the baselines. MGAN does not only use the dot attention mechanism and deep bidirectional LSTM, but also use a fine-grained attention mechanism to capture the word-level interaction between the aspect and context.

4.3 Ablation Study

To investigate the effectiveness of each component of our model, we perform comparison between the full model and its ablations. The result is shown in Table 4.

Table 4. Ablation experiments on three datasets

Models	Laptop		Restaurant		Twitter	
	Acc.	Macro-F1	Acc.	Macro-F1	Acc.	Macro-F1
SACA	**0.7633**	**0.7292**	**0.8205**	**0.7310**	**0.7268**	**0.7103**
- ELMo	0.7555	0.7148	0.8041	0.7124	0.7138	0.6984
- Char.	0.7601	0.7204	0.8189	0.7297	0.7221	0.7017
- Self-Attn.	0.7534	0.7017	0.8091	0.7188	0.7119	0.6914
- Co-Attn.	0.7392	0.6911	0.7828	0.7013	0.7064	0.6768
- Position	0.7609	0.7257	0.8123	0.7299	0.7108	0.6891

Effect of the embeddings. We perform ablation study on words and character embeddings. As expected, contextualized ELMo embeddings have a noticeable effect on each metric. Removing ELMo leads to ∼1 Acc. point drop on each dataset. Furthermore, after removing the character-level word embeddings, the performance degraded, slightly. It shows that the integration of multi-level information (*e.g.*, word and character level) is crucial for good performance.

Why self-attention? We conduct experiment to study the effect of self-attention mechanism. Firstly, we remove the self-attention and directly use hidden output of BiGRU to conduct experiment. The experimental result is shown in Table 4. After removing the self-attention layer, we find the performance degrade to 75.34 on Laptop dataset, 80.91 on Restaurant dataset and 71.19 on Twitter dataset, which verifies that the self-attention mechanism is effective for ABSA task.

Why co-attention? We compare the performance of SACA and SACA without co-attention layer in Table 4, we see that co-attention outperforms no co-attention method on three datasets. It shows that the interaction information between sentence and aspect is crucial for good performance.

Effect of position information. As for the position information, we removed position-aware attention and convolutional layer, thus retaining only max-pooling. It is worth noting that the performance is not much worse than SACA, especially on Laptop dataset and Restaurant dataset, which shows that the ability of position-aware attention is limited. We argue that it is because the importance of a context word is not only dependent on word order, but also on the information of context and aspect.

4.4 Case study

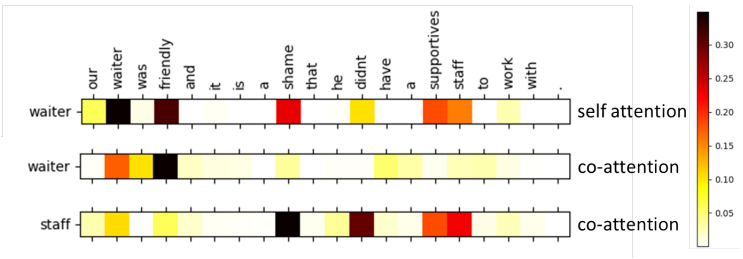

Fig. 2. Attention visualizations of an example sentence. The color depth indicates the importance degree of a word.

We take a sentence in restaurant dataset as example for illustrating the effectiveness of our proposed model. We visualize the weights of self-attention layer and co-attention layer. Fig. 2 shows the visualization results of the attention weights from self-attention and co-attention layer. The sentence in Fig. 2 is "*Our waiter was friendly and it is a shame that he didn't have a supportive staff to work with*". It has two aspects "*waiter*" and "*staff*", whose sentiment polarities are positive and negative, respectively. For the self-attention, we use the aspect "*waiter*" as query, visualize the scaled product attention weight. From the top

bar, we can find that the aspect "*waiter*" has more attention weight on "*friendly*", "*supportive*" and "*staff*". This phenomenon shows that the self-attention can not only learn the important context words for the aspect, but also can extract interactions between words, especially focus on neighboring words.

In addition, we evaluate the effect of co-attention mechanism and visualize the aspect to context attention weights. From the bottom two bars, we can observe that the co-attention can enforce the model to pay more attentions on the important words with respect to the aspect. For example, the words "*shame*", "*didn't*", and "*supportive*" which are the most relevant to sentiment polarity of *"staff"* has higher attention weights compared with other words.

5 Conclusion

In this paper, we present a new framework, term **SACA**, for aspect-based sentiment analysis. we re-examine the drawbacks of word representations for ABSA, to solve these issues, contextualized word representation and character-level word embeddings are integrated to word representations. Moreover, three novel attention mechanisms, namely self-attention, co-attention and position-aware attention mechanism have been introduced to our model. The self-attention captures the important information from a global perspective by considering the information of entire sentence. Co-attention mechanism captures the word-level interaction between aspect and context. Experimental results demonstrate the effectiveness of our approach on three datasets. The ablation studies show the efficacy of different modules.

Acknowledgements. This work was supported by the Ministry of Education of Humanities and Social Science project (No.19YJAZH128), and Guangdong Graduate Education Innovation project (No. 2018JGXM41).

References

1. Kiritchenko, S., Zhu, X., Cherry, C., Mohammad, S., Polosukhin, I.: NRC-Canada-2014: Detecting aspects and sentiment in customer reviews. In: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014) (2014)
2. Wagner, J., Arora, P., Cortes, S., Barman, U., Bogdanova, D., Foster, J., Tounsi, L.: Dcu: Aspect-based polarity classification for semeval task 4. In: Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014) (2014)
3. Bahdanau, D., Cho, K., Bengio, Y.: Neural Machine Translation by Jointly Learning to Align and Translate. arXiv preprint arXiv:1409.0473 (2014)
4. Wang, Y., Huang, M., Zhao, L.: Attention-based LSTM for aspect-level sentiment classification. In: Proceedings of EMNLP (2016)
5. Ma, D., Li, S., Zhang, X., Wang, H.: Interactive attention networks for aspect-level sentiment classification. In: Proceedings of IJCAI (2017)
6. Chen, P., Sun, Z., Bing, L., Yang, W.: Recurrent attention network on memory for aspect sentiment analysis. In: Proceedings of EMNLP (2017)

7. He, R., Lee, W.S., Ng, H.T., Dahlmeier, D.: Effective attention modeling for aspect-level sentiment classification. In: Proceedings of the 27th International Conference on Computational Linguistics (2018)
8. Tang, D., Qin, B., Liu, T.: Aspect level sentiment classification with deep memory network. In: Proceedings of EMNLP (2016b)
9. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. In: Proceedings of NAACL (2018)
10. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems (2017)
11. Vo, D.T., Zhang, Y.: Target-dependent twitter sentiment classification with rich automatic features. In: IJCAI (2015)
12. Tang, D., Qin, B., Feng, X., Liu, T.: Effective LSTMs for target-dependent sentiment classification. In: COLING (2016a)
13. Fan, F., Feng, Y., Zhao, D.: Multi-grained attention network for aspect-level sentiment classification. In: EMNLP (2018)
14. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems (2013)
15. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: EMNLP (2014)
16. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 (2014)
17. Pontiki, M., Galanis, D., Papageorgiou, H., Androutsopoulos, I., Manandhar, S.: SemEval-2014 task 4: Aspect based sentiment analysis. In: Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016) (2014)
18. Dong, L., Wei, F., Tan, C., Tang, D., Zhou, M., Xu, K.: daptive recursive neural network for target-dependent twitter sentiment classification. In: ACL (2014)
19. Kingma, D. P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980. (2014)
20. Majumder, N., Poria, S., Gelbukh, A., Akhtar, M.S., Cambria, E., Ekbal, A.: IARM: Inter-aspect relation modeling with memory networks in aspect-based sentiment analysis. In: EMNLP (2018)
21. Srivastava, R.K., Greff, K., Schmidhuber, J.: Highway network. arXiv preprint arXiv:1505.00387 (2015)