





Effective Self Attention Modeling for Aspect Based Sentiment Analysis

Ningning Cai^{1,2} , Can Ma¹ , Weiping Wang¹, and Dan Meng¹

¹ Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{cainingning, macan, wangweiping, mengdan}@iie.ac.cn
² University of Chinese Academy of Sciences, Beijing, China

Abstract. Aspect Based Sentiment Analysis is a type of fine-grained sentiment analysis. It is popular in both industry and academic communities, since it provides more detailed information on the user generated text in product reviews or social network. Therefore, we propose a novel framework based on neural network to determine the polarity of a review given a specific target. Not only the words close to the target but also the words far from the target determine the polarity of the review given a certain target, so we use self attention to solve the problem of long distance dependence. Briefly, we do multiple linear mapping on the review, do multiple attention and combine them to attend to the information from different representation sub-spaces. Besides, we use domain embedding to get close to the real word embedding in a certain domain, since the meaning of the same word may be different in different situation. Moreover, we use position embedding to underline the target and pay more attention to the words that are close to the target to get better performance on the task. We validate our model on four benchmarks, they are SemEval 2014 restaurant dataset, SemEval 2014 laptop dataset, SemEval 2015 restaurant dataset and SemEval 2016 restaurant dataset. The final results show that our model is effective and strong, which brings a 0.74% boost averagely based on the previous state-of-the-art work.

Keywords: Aspect Based Sentiment Analysis ·
Long Short-Term Memory (LSTM) · Attention

1 Introduction

Aspect Based Sentiment Analysis (ABSA) is a subtask of sentiment analysis. Instead of predicting the polarity of the overall sentence, it's proposed to predict the sentence polarity towards a given target. There are two subtasks [27], namely Aspect Category Sentiment Analysis (ACSA) and Aspect Term Sentiment Analysis (ATSA). The goal of ACSA is to predict the polarity with regard to a given target, which is one of some prepared and specific categories. And

This work is supported by the National Key Research and Development Program (Grant No. 2016YFB1000604).

the ATSA is to predict the polarity towards the given target, which is a sub sequence of the sentence. For example, given a sentence “*I bought a new camera. The picture quality is amazing but the battery life is too short*”, it’s ACSA if the target is “*price*” and it’s ATSA if the target is “*picture quality*”. Here, we mainly deal with the second task. As for the ATSA, if the target is “*picture quality*”, the expected sentiment polarity is positive as the sentence expresses positive emotion towards the target, but if the target is “*battery life*”, the true prediction should be negative. In other words, the polarity of a sentence may be opposite towards different targets. So the main challenge of ABSA is to find the words that actually determine the polarity towards a given target.

Now, we’ll introduce some core technique we used in this paper. LSTM has the remarkable capacity of modeling the sequence, so there are some previous works based on LSTM. [21] uses two LSTM to model the left and right sequence of the target. However, the key information could disappear if the key words are far from the target. Attention mechanism has been proven effective in many Natural Process Language task, such as machine translation [1]. Therefore, many great works that base on attention and LSTM make progress in dealing with the ABSA task. [25] builds up a variational attention layer on the top of LSTM, [19] stacks multiple attention layers and the experimental results show it is resultful. [2] does multiple attention operation and combines them with a non-linear method.

The self attention mechanism plays an important role in many tasks, such as [11, 17, 22]. In this paper, we propose a novel model with self attention which builds up a self attention layer on the top of the bi-LSTM layer. Specifically, we do multiple linear mapping on the input sentence, and do multiple attention operation on each of them, finally, we concatenate them.

Besides, we come up with an original multiple word embedding. As we all know, the same word may have different meaning in diverse situation, so either the word embedding. For example, “hot” in “hot dog” is totally different from “hot” in “Today is so hot” or “The girl is hot”. So apart from the general embedding that is trained from large corpus [12], we introduce the domain embedding, which is trained from a certain domain corpus. For example, if the ATSA task is about restaurant, then the domain embedding is trained from large restaurant corpus. Moreover, we introduce another novel word embedding, the position embedding. The position information is so important that it has been used with different methods in previous works [7]. In our paper, we use one-dimensional vector to represent it. The target is 0 and others are the distance from the given target. This way not only highlights the target phrase but also emphasizes the words close to the target.

We evaluate our model on four benchmarks, SemEval 2014 [15], containing the reviews of restaurant domain and laptop domain, SemEval 2015 restaurant dataset [14] and SemEval 2016 restaurant dataset [13]. The results prove that our model perform better than other baselines for all of the benchmarks, it gets competitive or even state-of-the-art results.

In general, our contributions are as follows: (i) introduce the domain embedding and firstly use position embedding in the embedding layer as far as we know; (ii) to our knowledge, we firstly use self attention in this area and come up with a novel framework; (iii) get the state-of-the-art results on four benchmarks.

The remainder of the paper is organized as follows. Section 2 introduces other related excellent work in this area and the differences among us. Section 3 introduces our model with detailed information. Section 4 is the details of the experiments. Finally, Sect. 5 is a further analysis of our model.

2 Related Work

There are many abundant and excellent works in the area of ABSA, which in literature is a fine-grained classification task [16]. The previous works are basically rule based or statistic based. [28] incorporates target dependent features and employs Support Vector Machine (SVM) to get comparable results. [3] employs probabilistic soft logic model to solve the problem. They [5, 9, 24] usually need expensive artificial features, such as n-grams, part-of-speech tags, lexicon dictionaries, dependency parser information and so on.

Since the neural network has the ability to capture features automatically through multiple hidden layers, there are more and more outstanding models based on neural network in this area. [23] extracts a rich set of automatic features through multiple embedding and multiple neural pooling function. [4] uses the dependency parsing results, regards the target word as the tree root and propagates the sentiment of the words from the tree bottom to the tree root node. However, the use of the dependency parser makes it not effective enough if the data is noisy like twitter data. [27] comes up with a model Gated Convolutional network with Aspect Embedding (GCAE), which is a pure Convolution Neural Network and uses gating mechanism to assign different weights to the words. [19] uses two LSTM to model the sequence from the beginning and the tail to the target word. And it has to be noted that if the decisive words are far from the target, the model may fail.

Furthermore, attention based LSTM has gained a lot of attention due to their ability to capture the importance of the words. [21] stacks multiple attention layer and gets competitive results. [25] comes up with a variant of LSTM with attention, they add the target embedding to each of the hidden units. [2] also adopts multiple attention layers and combines the outputs with a Recurrent Neural Network (RNN) model. [7] incorporates syntactic information into the attention mechanism. We also use self attention based on bi-LSTM. The self attention does multiple linear mapping on the input sentence, does multiple attention and combines them. Besides the self attention, we also use domain embedding and position embedding. The former has been proved effective in extraction task [26], the latter is usually used in the attention layer and is computed by dependency parser [7], here we use it in the embedding layer in a simple but effective way.

3 Model

The architecture of our model is shown in Fig. 1, which consists of four modules, word embedding module, bi-LSTM module, self attention module and softmax output module. ATSA aims to determine the sentiment polarity of a sentence s towards a given target word or phrase a , a sub-sequence of s .

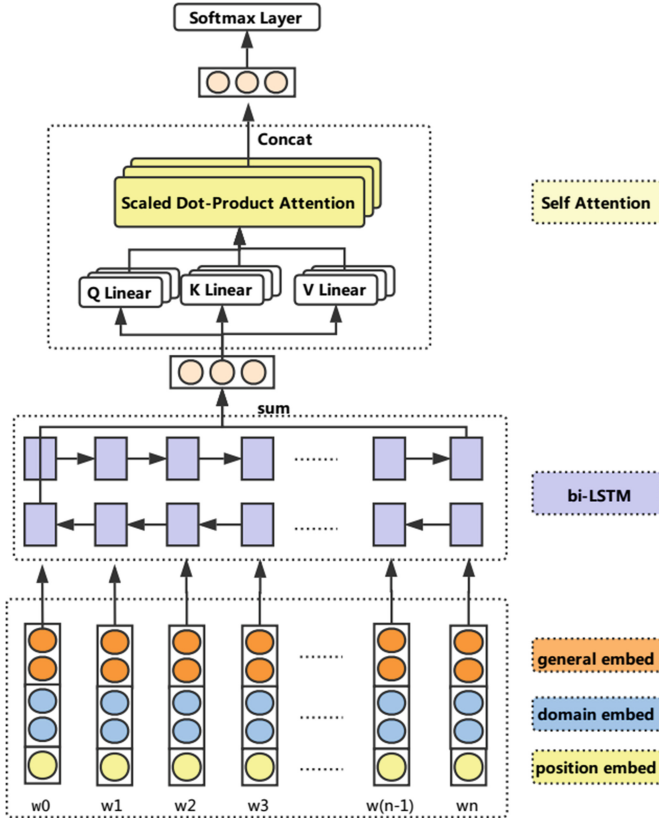


Fig. 1. The architecture of our model.

3.1 Word Embedding

The input is a sentence $s = (w_0, w_1, w_2, \dots, w_n)$, which consists of a given target $a = (a_0, a_1, \dots, a_m)$. Each of the above word w_i is presented as a continuous and dense numeric vector e_{w_i} from a look up table named word-embedding-matrix $E \in \mathbb{R}^{V \times d}$, where V is the vocabulary size and d is the word embedding dimension. The word embedding concatenate three different components, which are general embedding $E^g \in \mathbb{R}^{V \times d_g}$, domain embedding $E^d \in \mathbb{R}^{V \times d_d}$ and position embedding $E^p \in \mathbb{R}^{V \times d_p}$. Usually, other related works use general embedding only, but we introduce the other two to improve the performance.

General Embedding. The general embedding matrix E^g is pre-trained from a large corpus irrelevant to the specific task, such as glove.840B.300d [12].

Domain Embedding. The domain embedding matrix E^d is pretrained from a corpus relevant to the specific task. For example, if the ABSA task is about restaurant, then we pretrain word embedding from large restaurant corpus like Yelp Dataset [20]. And the reason we introduce it is vectors trained from out-of-domain corpus can't express their true meaning properly. For instance, "hot" in "hot dog" would be close to warm or something about weather and "dog" would be close to something about animals. However, this kind of expression is far from its true meaning that unexpectedly it turns out to be a kind of food.

Position Embedding. Intuitively, not all words are equal important to classify the polarity of the sentence given a target, usually words appear near the target and words have translation relation need more attention. We use a one-dimensional vector to represent each word w_i . The number is the distance from the target. Suppose there is a sentence that "I love [the hot dog]_{target} very much", then in our paper its position embedding is [2 1 0 0 0 1 2]. We mark the target with 0 to distinguish the target from others and other distances characterize the different importance to the classification task.

3.2 bi-LSTM Layer

Long Short-Term Memory (LSTM) [6] is a type of variational RNN in order to overcome the vanishing gradient problem, so it's a powerful tool to model the long sequence. bi-LSTM can capture more information than LSTM, since both forward information and backward information can be used to infer. We use bi-LSTM to process the input sentence in opposite direction and sum the last hidden vectors as the output. The $h_t = LSTM(h_{t-1}, e_{wt})$ is calculated as follows, where W is the weight matrix and b is the bias:

$$f_t = \sigma(W_f \times [h_{t-1}, e_{wt}] + b_f) \quad (1)$$

$$i_t = \sigma(W_i \times [h_{t-1}, e_{wt}] + b_i) \quad (2)$$

$$o_t = \sigma(W_o \times [h_{t-1}, e_{wt}] + b_o) \quad (3)$$

$$\tilde{c}_t = \tanh(W_c \times [h_{t-1}, e_{wt}] + b_c) \quad (4)$$

$$c_t = f_t \times c_{t-1} + i_t \times \tilde{c}_t \quad (5)$$

$$h_t = o_t \times \tanh(c_t) \quad (6)$$

The bi-LSTM academic description is just like this:

$$\vec{h}_t = LSTM(\vec{h}_{t-1}, e_{wt}) \quad (7)$$

$$\overleftarrow{h}_t = LSTM(\overleftarrow{h}_{t-1}, e_{wt}) \quad (8)$$

$$output = \vec{h}_t + \overleftarrow{h}_t \quad (9)$$

where e_{wt} is the embedding vector of the word w_t which is the t th of the input sentence s and h_t is the corresponding hidden state.

3.3 Self Attention

Self-attention is a special attention mechanism to compute the representation of a sentence. It has been proved effective in many Natural Language Processing (NLP) tasks, such as Semantic Role Labeling [18], Machining Translation [22] and other tasks [11, 17]. In this section, firstly, we introduce the self-attention, then we discuss its advantages.

Scaled Dot-Product Attention. Given a query matrix $Q \in \mathbb{R}^{n \times d}$, key matrix $K \in \mathbb{R}^{n \times d}$ and value matrix $V \in \mathbb{R}^{n \times d}$, we calculate the scaled dot-product attention *head* as follows. Here, n means we pack n queries, keys or values together into matrix Q , K and V , and d is the dimension of them:

$$head(Q, K, V) = softmax\left(\frac{QK}{\sqrt{d}}V\right) \quad (10)$$

The divisor \sqrt{d} is pushing the softmax function into regions where it has extremely small gradients [22].

Multi-head Attention. The mechanism firstly does linear mapping on the input matrices Q , K and V , repeats h times and then concatenates the results as output m . The h parallel operations allow the model to jointly attend to information from different representation sub-spaces.

$$m = concat(head_1, head_2, \dots, head_h)W^m$$

$$where\ head_i = head(QW^q, KW^k, VW^v) \quad (11)$$

In our paper, the input Q , K , V are all the output of the bi-LSTM layer. The self attention can capture dependencies even if the distance of the words are too far. The distance of each two words are 1 while it can be n (the sequence length) in RNN architecture. Also, it's highly parallel while RNN is not. At the same time, Features it captures are more abundant than CNN since CNN uses the fixed window size.

3.4 Softmax Layer

The ABSA is a three classification task whose label is positive, negative or neural. The self attention layer's output m is the representation of the given sentence, and we feed it into a softmax layer to predict the probability distribution p over sentiment label, where W_o is the weight matrix and b_o is the bias:

$$p = softmax(W_o m + b_o) \quad (12)$$

The training object is minimizing cross-entropy function:

$$loss = \sum_{i \in C} \log p_i(t_i) \quad (13)$$

where C is the training corpus, p_i is the prediction label while t_i is the real label.

4 Experiments

4.1 Datasets and Preparations

We validate our model on four benchmarks, they are SemEval2014 [15], containing two datasets, SemEval2015 [14] and SemEval2016 [13]. The statistics of them is shown in Table 1. Following the previous work [8], we also remove the data with *conflicting* label.

Table 1. The positive, negative and neural examples statistics of Semval Datasets

Datasets	Train			Test		
	Pos	Neg	Neu	Pos	Neg	Neu
SemEval 2014_restaurant	2164	805	633	728	196	196
SemEval 2014_laptop	987	866	460	341	128	169
SemEval 2015_restaurant	1178	382	50	439	328	35
SemEval 2016_restaurant	1620	709	88	597	190	38

4.2 Evaluation Metric

We use the accuracy metric **acc** to evaluate our model. The method is:

$$acc = \frac{TP}{TP + FP} \quad (14)$$

where TP is true positive and FP is false positive.

4.3 Hyper-parameters Settings

In all of our experiments, 300-dimension E^g is initialed by Glove [12], 100-dimension E^d is trained by fasttext¹ with yelp [20] corpus and the Amazon Electronics dataset [10]. We randomly pick up 20% of training data as development data to keep the best parameters. The optimizer is Root Mean Square Prop (RMSProp) with initial learning rate 0.001. The dimension of the bi-LSTM is 400. The epoch is 25 and the mini batch is 32. We use dropout with 0.5 and early-stopping to prevent from overfitting. The number of multi-head h is 16.

4.4 Model Comparison

We compare some traditional models with our model, they are as follows.

SVM with labor features [9] is a typical statistic model. The SVM is trained with a lot of labor features, including n-grams, POS labels and large-scale lexicon dictionaries. We compare with the reported results on SemEval2014.

¹ ref: <https://github.com/facebookresearch/fastText>.

Table 2. Average accuracies over 3 runs with random initialization. The best results are in bold.

Model	SemEval 2014_res	SemEval 2014_lt	SemEval 2015_res	SemEval 2016_res
SVM	80.16	70.49	NA	NA
LSTM	75.27	66.55	74.06	82.39
LSTM-ATT	75.74	67.55	76.14	83.48
TD-LSTM	75.37	68.25	76.38	82.16
ATAE-LSTM	78.60	68.88	78.48	83.77
RAM	78.48	72.08	79.98	83.88
PRET+MULT	79.11	71.15	81.30	85.58
SA-LSTM (OURS)	80.15	72.42	81.92	85.62

LSTM [6] We build up a LSTM layer on the word embedding layer, and the output is the average of the hidden states.

LSTM + attention (ATT) Based on the above LSTM, we add an attention layer on the top of the LSTM layer. Briefly, we calculate the weight α of each hidden state h and multiply them as the sentence representation. The weight α is described with the following equation:

$$target = \frac{1}{m} \sum_{i=1}^m e_{ai} \quad (15)$$

$$d_i = \tanh(h_i, target) \quad (16)$$

$$\alpha_i = \frac{\exp(d_i)}{\sum_{j=1}^n \exp(d_j)} \quad (17)$$

Target-dependent LSTM (TD-LSTM) [21] They use one LSTM to model the sequence from the beginning to the target and another LSTM to model the sequence from the end to the target, then they combine the results as the sentence representation.

Attention-based LSTM with Aspect Embedding (ATAE-LSTM) [25] is a variant of LSTM+ATT, they add target embedding vector to each of the LSTM hidden states.

Recurrent Attention Network on Memory (RAM) [2] uses LSTM and multiple attention. Briefly, they use multiple attention operation and combines them with a RNN as the sentence representation.

Pre-train + Multi-task learning (PRET+MULT) [8] use pre-train and multi-task learning to get better performance. They use another document level sentiment analysis task as the auxiliary task.

The results are shown in Table 2, and it’s the average value over three times with random initialization. The results indicate that our model is effective and strong in four different benchmarks. More abundant analyses will be in next section.

4.5 Analysis

Table 2 indicates that we can gain a lot from multi-embedding and self attention. Our model bring a 0.74% boost averagely based on the previous state-of-the-art work. We can find that the improvement of th previous two dataset are better than the SemEval2015_res and SemEval2016_res dataset, and we think it’s because the problem of the label imbalance is less serious on the previous two dataset. For further verification, we do more experiments whose results are shown in Fig. 2.

To validate the effectiveness of the word embedding layer, (1) we remove the domain embedding from the model, the experiment result decreases by 0.60%–1.62%, the average is 1.05%. (2) we remove the position embedding from the model, the experiment result decreases by 1.74%–2.74%, the average is 2.18%. On the whole, the position embedding plays more important role than the domain embedding in the ATSA task. Intuitively, the phenomenon is reasonable because the position embedding not only stress the target information but also pay more attention to the words close to the target.

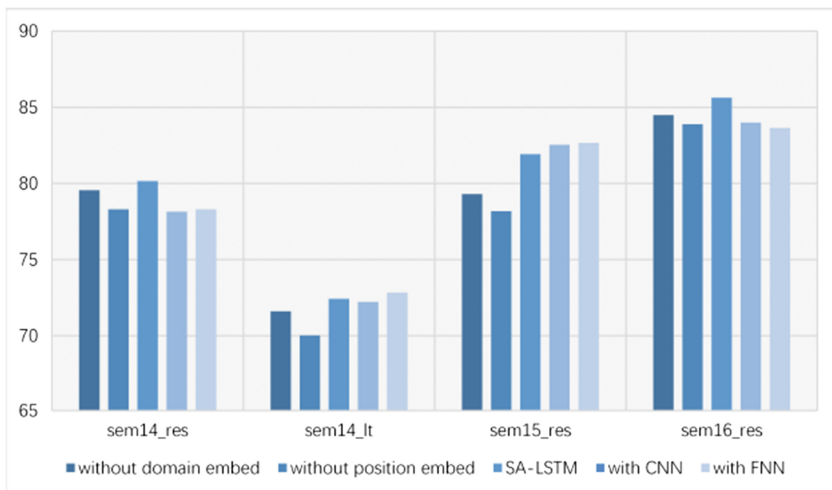


Fig. 2. More experiments to valid the effectiveness of the model. The five setting from the left to the right is: (1) without domain embedding in the embedding layer, (2) without position embedding in the embedding layer, (3) our model SA-LSTM, (4) replace the bi-LSTM with CNN in the bi-LSTM layer, (5) replace the bi-LSTM with FNN in the bi-LSTM layer. The y axis is the accuracy on the four datasets.

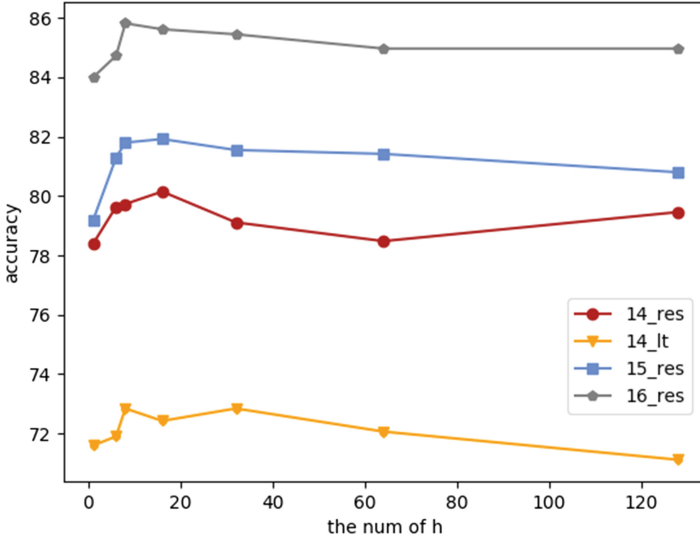


Fig. 3. The influence of the multi-head nums in the self attention layer. The y axis is the accuracy on the four datasets.

To approve the potential of the bi-LSTM layer, (1) we replace the second layer bi-LSTM with Convolutional Neural Networks (CNN) inspired by [27], the computation is as follows:

$$a_i = (X_{i,i+k} W_1 + b_1) \quad (18)$$

$$b_i = \text{sigmoid}(X_{i,i+k} W_2 + b_2) \quad (19)$$

$$\text{output}_i = a_i \times b_i \quad (20)$$

where k is the window size, here we set it with 3 and X is the input sentence after embedding. The result shows it isn't good as bi-LSTM, which decreases by 1.2% averagely on three benchmarks but increases by 0.62% on one benchmark.

(2) we replace the second layer bi-LSTM with FNN (Forward Neural Network), the computation is as follows:

$$\text{output} = \text{relu}(X W + b_1) \quad (21)$$

The FNN is so simple but perform well, which follows Occam's razor principle that simple is the best. It decrease by over 2% on two benchmarks but increases by about 0.5% on two benchmarks.

Additionally, In order to get the influence of the factor multi-head number h , we draw the Fig. 3. The figure pinpoints that it's not the more the better, most benchmarks get their best performance when h is 16. However, SemEval2014.It dataset gets its best performance when h is 32.

5 Conclusion

To our knowledge, our work is the first attempt to use the domain and position embedding in the embedding layer and the first attempt to use self attention in the ABSA area. We have validated the effectiveness of our model and we get competitive or even the state-of-the-art results on four benchmarks. In the future, we'll attempt to model sentence and target separately with self attention to get better performance and focus on the problem of label imbalance. Besides, we may also try other position embedding strategies to give the important words more attention.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. *Computer Science* (2014)
2. Chen, P., Sun, Z., Bing, L., Yang, W.: Recurrent attention network on memory for aspect sentiment analysis. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 452–461 (2017)
3. Deng, L., Wiebe, J.: Joint prediction for entity/event-level sentiment analysis using probabilistic soft logic models. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 179–189 (2015)
4. Dong, L., Wei, F., Tan, C., Tang, D., Zhou, M., Xu, K.: Adaptive recursive neural network for target-dependent twitter sentiment classification. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, vol. 2, pp. 49–54 (2014)
5. Ganapathibhotla, M., Liu, B.: Mining opinions in comparative sentences. In: *Proceedings of the 22nd International Conference on Computational Linguistics*, vol. 1, pp. 241–248. Association for Computational Linguistics (2008)
6. Gers, F.A., Schmidhuber, J., Cummins, F.: Learning to forget: continual prediction with LSTM (1999)
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*, pp. 1121–1131 (2018)
8. He, R., Lee, W.S., Ng, H.T., Dahlmeier, D.: Exploiting document knowledge for aspect-level sentiment classification. *arXiv preprint [arXiv:1806.04346](https://arxiv.org/abs/1806.04346)* (2018)
9. Kiritchenko, S., Zhu, X., Cherry, C., Mohammad, S.: NRC-Canada-2014: detecting aspects and sentiment in customer reviews. In: *International Workshop on Semantic Evaluation*, pp. 437–442 (2014)
10. Mcauley, J., Targett, C., Shi, Q., Hengel, A.V.D.: Image-based recommendations on styles and substitutes (2015)
11. Paulus, R., Xiong, C., Socher, R.: A deep reinforced model for abstractive summarization. *arXiv preprint [arXiv:1705.04304](https://arxiv.org/abs/1705.04304)* (2017)
12. Pennington, J., Socher, R., Manning, C.: Glove: global vectors for word representation. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543 (2014)
13. Pontiki, M., et al.: Semeval-2016 task 5: aspect based sentiment analysis. In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pp. 19–30 (2016)

14. Pontiki, M., Galanis, D., Papageorgiou, H., Manandhar, S., Androutsopoulos, I.: Semeval-2015 task 12: aspect based sentiment analysis. In: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pp. 486–495 (2015)
15. Pontiki, M., Galanis, D., Pavlopoulos, J., Papageorgiou, H., Androutsopoulos, I., Manandhar, S.: Semeval-2014 task 4: aspect based sentiment analysis. In: Proceedings of International Workshop on Semantic Evaluation, pp. 27–35 (2014)
16. Rojas-Barahona, L.M.: Deep learning for sentiment analysis. *Lang. Linguist. Compass* **10**(12), 701–719 (2016)
17. Shen, T., Zhou, T., Long, G., Jiang, J., Pan, S., Zhang, C.: Disan: directional self-attention network for RNN/CNN-free language understanding. arXiv preprint [arXiv:1709.04696](https://arxiv.org/abs/1709.04696) (2017)
18. Tan, Z., Wang, M., Xie, J., Chen, Y., Shi, X.: Deep semantic role labeling with self-attention. arXiv preprint [arXiv:1712.01586](https://arxiv.org/abs/1712.01586) (2017)
19. Tang, D., Qin, B., Feng, X., Liu, T.: Effective LSTMs for target-dependent sentiment classification. arXiv preprint [arXiv:1512.01100](https://arxiv.org/abs/1512.01100) (2015)
20. Tang, D., Qin, B., Liu, T.: Document modeling with gated recurrent neural network for sentiment classification. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1422–1432 (2015)
21. Tang, D., Qin, B., Liu, T.: Aspect level sentiment classification with deep memory network. arXiv preprint [arXiv:1605.08900](https://arxiv.org/abs/1605.08900) (2016)
22. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)
23. Vo, D.T., Zhang, Y.: Target-dependent twitter sentiment classification with rich automatic features. In: International Conference on Artificial Intelligence, pp. 1347–1353 (2015)
24. Wagner, J., et al.: DCU: aspect-based polarity classification for semeval task 4. In: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pp. 223–229 (2014)
25. Wang, Y., Huang, M., Zhao, L., et al.: Attention-based LSTM for aspect-level sentiment classification. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 606–615 (2016)
26. Xu, H., Liu, B., Shu, L., Yu, P.S.: Double embeddings and CNN-based sequence labeling for aspect extraction. arXiv preprint [arXiv:1805.04601](https://arxiv.org/abs/1805.04601) (2018)
27. Xue, W., Li, T.: Aspect based sentiment analysis with gated convolutional networks. arXiv preprint [arXiv:1805.07043](https://arxiv.org/abs/1805.07043) (2018)
28. Zhou, M.: Target-dependent twitter sentiment classification. In: Proceedings of Annual Meeting of the Association for Computational Linguistics Human Language Technologies, vol. 1, pp. 151–160 (2011)