Dependency Parsing and Attention Network for Aspect-Level Sentiment Classification

Zhifan Ouyang and Jindian Su(✉)

School of Computer Science and Engineering,
South China University of Technology, Guangzhou, China
csouzhi@mail.scut.edu.cn, sujd@scut.edu.cn

Abstract. Aspect-level sentiment classification aims to determine the sentiment polarity of the sentence towards the aspect. The key element of this task is to characterize the relationship between the aspect and the contexts. Some recent attention-based neural network methods regard the aspect as the attention calculation goal, so they can learn the association between aspect and contexts directly. However, the above attention model simply uses the word embedding to represent the aspect, it fails to make a further improvement on the performance of aspect sentiment classification. To solve this problem, this paper proposes a dependency subtree attention network (DSAN) model. The DSAN model firstly extracts the dependency subtree that contains the descriptive information of the aspect based on the dependency tree of the sentence, and then utilizes a bidirectional GRU network to generate an accurate aspect representation, and uses the dot-product attention function for the dependency subtree aspect representation, which finally yields the appropriate attention weights. The experimental results on SemEval 2014 Datasets demonstrate the effectiveness of the DSAN model.

Keywords: Aspect-level sentiment classification · Attention network
Dependency tree

1 Introduction

Aspect-level sentiment classification is a fine-grained task in the field of sentiment analysis [13], which aims to identify the sentiment expressions for aspects in their contexts. This task can provide more detailed and in-depth sentiment analysis results, which has been getting much attention recently. For example, given a sentence *"Air has higher resolution, but the fonts are small."*, the polarity is positive for the aspect *"resolution"*, and negative for the aspect *"fonts"*.

Because the sentiment polarity of an aspect needs to consider both the aspect and the contexts, the key point is how to characterize the relationship between the aspect and the contexts [19]. Dependency parsing plays a very important role in the aspect-level sentiment classification task. In some previous work, the

M. Zhang et al. (Eds.): NLPCC 2018, LNAI 11108, pp. 391–403, 2018.
https://doi.org/10.1007/978-3-319-99495-6_33

dependency tree is used to extract aspect-related features to build sentiment classifiers in traditional machine learning methods [7], or to establish aspect-specific recursive structure used for the input in Recursive Neural Network methods [4,11].

Since attention mechanism can help to enforce a model to learn the task-related parts of a sentence [1], some works exploit this advantage and achieve superior performance for aspect-level sentiment classification, i.e. [2,3,15,17]. The attention-based models regard the aspect as the attention calculation goal, which enable the model to learn the association between aspect and its contexts directly. Usually, different aspects in the same sentence might have different attention weights. Despite of the advantages of attention mechanism, previous models simply use the word embedding of the aspect to represent the aspect and calculate the corresponding attention weights, which as a result might lose a lot of aspect information and the aspect representations are not accurate enough, so that the attention model fails to learn the appropriate attention weights for each aspect.

The generation of an accurate aspect representation for each aspect becomes an important factor to make further improvements on the performance of the attention models for aspect-level sentiment classification. Similar to modeling contextual information, utilizing a Recurrent Neural Network (RNN) to model the aspect and generate aspect representations is a very worthwhile try, which is the same as to Ma et al. [9]. However, it is insufficient to simply model the aspect in the form of a noun or a noun phrase.

From the perspective of sentiment expression, we found that when people express their sentiment about some specific target, they tend to use adjectives or adverbs to describe the target and express their inner feelings. These modifiers are closely related with the target, and can form the accurate descriptions of the aspect in the sentence. By using dependency parsing, we can obviously see that these modifiers are generally subject to the aspect. Therefore, we can extract a dependency subtree of the aspect based on the dependency tree of the sentence. For example, we can extract two dependency subtrees from the sentence *"Air has higher resolution, but the fonts are small."*, as shown in Fig. 1, each of which is

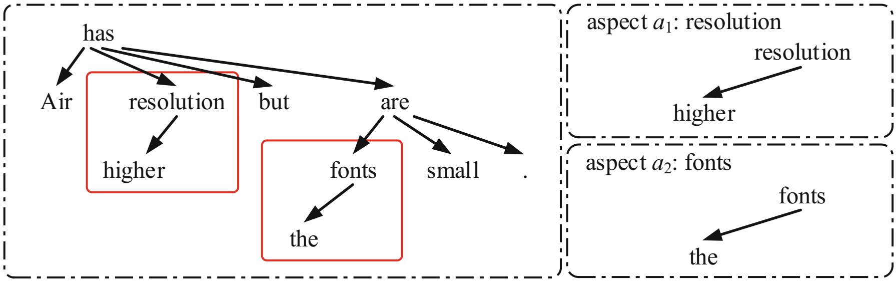

Fig. 1. Dependency parsing tree for sentence: *"Air has higher resolution, but the fonts are small."*, and the dependency subtree of aspect a_1 *"resolution"* and aspect a_2 *"fonts"*.

also a sub-sentence of the sentence and includes some context information about the aspect. So, we can try to model the aspect sub-sentence by RNN networks, and use it to denote the aspect instead of a noun or a noun phrase.

Motivated by the above intuition, we propose a dependency subtree attention network (DSAN) model, which is based on dependency parsing and attention mechanism. DSAN utilizes gated recurrent unit (GRU) to separately modelling the aspect sub-sentence and the contexts, and uses the attention mechanism to generate aspect-related sentiment features based on aspect sub-sentence. We have evaluated our model on Laptop and Restaurant datasets from SemEval 2014 [13]. Experimental results show that our DSAN model achieves the comparable state-of-the-art performance for aspect-level sentiment classification.

2 Model

In this section, we describe the dependency subtree attention network (DSAN) model for aspect-level sentiment classification. The architecture of DSAN model is shown in Fig. 2.

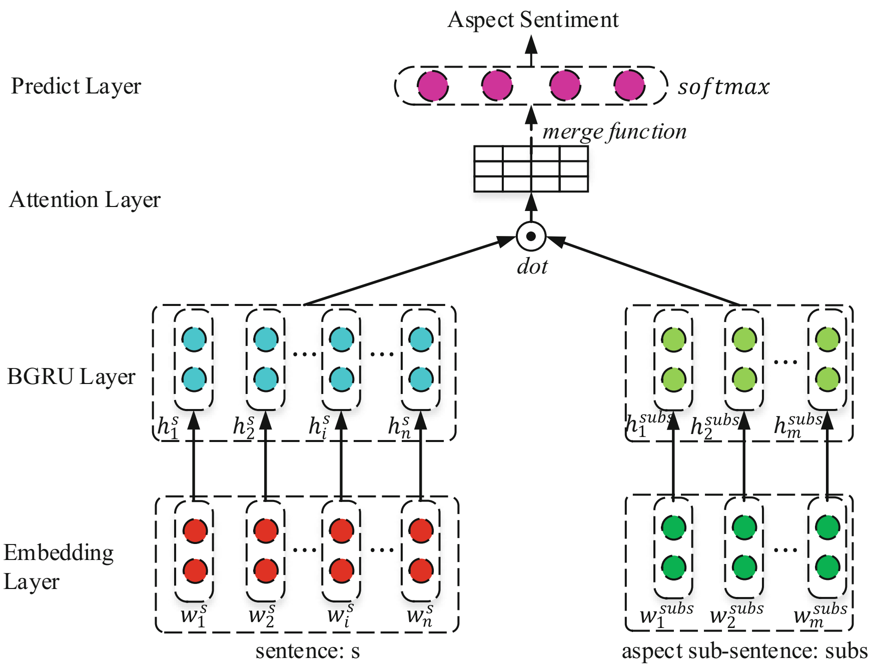

Fig. 2. The architecture of DSAN model

2.1 Embedding Layer

Let $L \in \mathbb{R}^{d_w \times |V|}$ be a word embedding look-up table which is usually trained on an external large corpus [10,12], d_w be the dimensions of word vectors and $|V|$ be the size of the vocabulary. Given a sentence $s = \{w_1^s, w_2^s, ..., w_i^s, ..., w_n^s\}$ and the dependency subtree of an aspect (sub-sentence) $subs = \{w_1^{subs}, w_2^{subs}, ..., w_i^{subs}, ..., w_m^{subs}\}$, the embedding layer project each word into a low dimensional, continuous and real-valued vector, denoted as $X^s = [x_1^s, x_2^s, ..., x_i^s, ..., x_n^s]$ where $x_i^s \in \mathbb{R}^{d_w}$ represents w_i^s, and $X^{subs} = [x_1^{subs}, x_2^{subs}, ..., x_i^{subs}, ..., x_n^{subs}]$ where $x_i^{subs} \in \mathbb{R}^{d_w}$ represents w_i^{subs}.

2.2 BGRU Layer

RNN has already demonstrated its superior performance on variable-length sequences modeling. It can capture long-term dependency information of words and is popularly used in the area of sentiment analysis [18]. In this paper, we use bidirectional GRU (BGRU) to separately model the contexts and the aspect. Denote the hidden states of the forward GRU at time step i as $\overrightarrow{h_i} = \overrightarrow{GRU}(x_i, \overrightarrow{h_{i-1}})$ and the backward GRU as $\overleftarrow{h_i} = \overleftarrow{GRU}(x_i, \overleftarrow{h_{i-1}})$, and $h_i = [\overrightarrow{h_i}; \overleftarrow{h_i}]$ as the output of BGRU at time step i. Then, the output of the contexts and the aspect modeled by BGRU can be defined as:

$$H^s = BGRU^s(X^s) \tag{1}$$

$$H^{subs} = BGRU^{subs}(X^{subs}) \tag{2}$$

where the hidden states $H^s = [h_1^s, h_2^s, ..., h_i^s, ..., h_n^s]$, $h_i^s \in \mathbb{R}^{2d}$ is the representations for the contexts, and the hidden states $H^{subs} = [h_1^{subs}, h_2^{subs}, ..., h_i^{subs}, ..., h_m^{subs}]$, $h_i^{subs} \in \mathbb{R}^{2d}$ is the representations for the aspect.

2.3 Position Weight

The position of the aspect helps the model to distinguish different aspects in the same sentence, and the sentiment expression of the aspect is close to the aspect in the contexts. Therefore, we bring the positional information of the aspect into consideration in the form of position weights during attention calculation. We define the distance as the path of the aspect and context word in the dependency tree. If the aspect is a phrase, then the distance will be simply calculated with the last word in the aspect phrase. The calculation formula of the weight for the context word w_i is:

$$l_i = 1 - \frac{dist_{i,a}}{2dist_{max}} \tag{3}$$

where $dist_{i,a}$ denotes the distance of context word w_i and the aspect a, $dist_{max}$ denotes the max distance of context words in the input sentence s. The range of the position weight is limited to $[0.5, 1]$.

Based on Eq. 3, we can get an aspect-customized hidden states of the sentence s, denoted as $E^s = [e_1^s, e_2^s, ..., e_i^s, ..., e_n^s]$, where $e_i^s = l_i \cdot h_i^s \in \mathbb{R}^{2d}$. The position weights can give higher weight to the context word which is close to the aspect. This can help the model to predict the sentiment of different aspects flexibly and prevent the model from being misled by a strong unrelated sentiment expression.

2.4 Attention Layer

The goal of the attention layer is to allocate appropriate attention weights to the words of the sentence according to the aspect. The attention layer generates aspect-related sentiment features. We employ the attention function which is similarly as Vaswani et al. [16], and the attention weights calculation is based on the hidden states H^{subs} of sub-sentence and aspect-customized hidden states E^s,

$$Q = relu(W_1 H^{subs}) \tag{4}$$

$$K = relu(W_2 E^s) \tag{5}$$

$$Score = \frac{K^T Q}{\sqrt{d_k}} \tag{6}$$

where $W_1, W_2 \in \mathbb{R}^{d_k \times 2d}$ are linear transfer parameters, $Score \in \mathbb{R}^{n \times m}$ is an attention score matrix. $Score_{i,j}$ represents the attention score of the word w_i in the sentence s and the word w_j in the sub-sentence $subs$. In practice, we found that adding rectified linear unit (relu) activation function to filter out negative values can yield a more stable performance. In order to merge the attention contributions to each word of the sub-sentence $subs$, we define a column merging function over score matrix $Score$,

$$\alpha = \begin{cases} softmax\left(\sum_{j=1}^{m} Score_j\right), & \text{if } mode = sum; \\ softmax\left(\frac{1}{m}\sum_{j=1}^{m} Score_j\right), & \text{if } mode = mean; \end{cases} \tag{7}$$

where m is the length of the sub-sentence $subs$, α is the final attention weights of context words. In this work, the merging function includes two different types, sum and $mean$.

After getting the attention weights, we can calculate the final aspect-related sentiment representations r as follows,

$$V = H^s + W_3 X^s \tag{8}$$

$$r = V \cdot \alpha \tag{9}$$

where $W_3 \in \mathbb{R}^{2d \times d_w}$ is a linear transfer parameter for the context word embedding sequence X^s. $V \in \mathbb{R}^{2d \times n}$ is the cumulative result of H^s and X^s, which can be viewed as a key-value memory network, whose keys and values are K and V [5] respectively.

2.5 Sentiment Predict Layer

Finally, we concatenate the last hidden states of BGRU of the sentence s to represent the sentence, and use a nonlinear transfer to get the final representations of a sentence after given an aspect,

$$r^s = [\overrightarrow{h^s_n}; \overleftarrow{h^s_1}] \tag{10}$$

$$h^* = relu\,(W_4 r + W_5 r^s + b_4) \tag{11}$$

where $h^* \in \mathbb{R}^{d_r}$, $W_4, W_5 \in \mathbb{R}^{d_r \times 2d}$ and $b_4 \in \mathbb{R}^{d_r}$ are the parameters of nonlinear layer. Then, we feed the representation h^* into a $softmax$ layer to predict the aspect sentiment polarity.

2.6 Model Training

We use the cross entropy as the objective function, and plus an L_2 regularization term to prevent overfitting,

$$J = \sum_{(x,y)\in D} \sum_{c \in C} P^g_c(x,y) log P_c(x,y;\theta) + \lambda \| \theta \|^2 \tag{12}$$

where C is the sentiment category set, D is the collection of training instance, $P^g(x,y)$ is a one-hot vector that indicates the true sentiment of aspect, $P(x,y;\theta)$ is the predicted sentiment probability of the model, θ is the parameter set, and λ is the regularization weight. We adapt the ADAM method to update parameters [8].

3 Experiment

3.1 Datasets and Settings

We conduct experiments on two datasets from SemEval task 4 [13], reviews of laptop and restaurant domain respectively. Each aspect with reviews is labeled with three sentiment polarities: positive, negative and neutral. Table 1 shows the final statistics of two datasets.

Table 1. Statistics of two datasets.

DataSet	Positive	Neutral	Negative
Laptop-train	987	460	866
Laptop-test	341	169	128
Restaurant-train	2164	633	805
Restaurant-test	728	196	196

We implement the DSAN model with Keras[1] and TensorFlow[2], and use spaCy[3] to parse the structures of sentences. We use the pre-trained GloVe word embeddings [12] for our experiments. In addition, we use the Amazon electronic dataset [6] for laptop domain, and the Yelp Challenge dataset[4] for restaurant domain to train 300-dimension word embeddings with the Skipgram[5] training method. All parameters except word embeddings are initialized with random uniform distribution $U(-0.05, 0.05)$. The hidden size is set to 120 for both two BGRU and 100 for the nonlinear layer. To prevent overfitting, we set regularization weight λ to be 0.012, and the dropout rate of two BGRU to be 0.5. Evaluation metrics are Accuracy and Macro-F1 because the datasets are unbalanced.

3.2 Experimental Results

In order to evaluate our DSAN model, we compare it with the following methods: **TD-LSTM**[14], **MemNet**[15], **RAM**[2], **ATAE-LSTM**[17], **IAN**[9].

Tables 2 and 3 show the experiment results of our model compared with other related models. We denote our models as **DSAN-sum**, **DSAN-mean**, which means using *sum* and *mean* mode respectively. In Table 2, all models use pre-trained Glove 300-dimension word embeddings, The difference is that the first group whose vocabulary size is 1.9M and the second one is 2.2M. In Table 3, all models use the skipgram word embeddings that are trained on domain-specific corpus.

From Tables 2 and 3, we can conclude:

(1) Our DSAN model achieves the comparable results with RAM method on both Laptop and Restaurant datasets, exceeding other four benchmark methods except the RAM, which is the state-of-the-art method for aspect-level sentiment classification. Compared with the RAM method, the two merge modes of the DSAN model can achieve better performance than the RAM on the Restaurant dataset, and its performance is slightly lower than the RAM on the Laptop dataset. The biggest difference between the RAM model and the DSAN model is that the RAM implements multiply attention mechanisms based on the GRU network, so it can catch information about different important parts of a sentence by attention layers, and combine the result of each attention layer in a non-linear manner. But RAM still simply uses the word embedding to represent the aspect. The DSAN model utilizes the bidirectional GRU to model the sub-sentence of the aspect that containing the descriptive information of aspect, so the aspect can be represented more accurately than using the way of word embeddings. Therefore, even

[1] https://keras.io.
[2] https://www.tensorflow.org.
[3] https://spacy.io/.
[4] https://www.yelp.com/dataset.
[5] https://radimrehurek.com/gensim/models/word2vec.html.

Table 2. Results of different methods on Laptop and Restaurant datasets. The results with '*' are retrieved from RAM paper, and the results with '$\diamond$' are retrieved from the papers of compared methods. Best results in each group are in bold.

Word embeddings	Model	Laptop		Restaurant	
		Accuracy	Macro-F1	Accuracy	Macro-F1
Glove (1.9M vocabulary size)	TD-LSTM	0.7183*	0.6843*	0.7800*	0.6673*
	MemNet	0.7033*	0.6409*	0.7816*	0.6583*
	RAM	**0.7449***	**0.7135***	0.8023*	0.7080*
	DSAN-sum	0.7273	0.6878	0.8071	0.7238
	DSAN-mean	0.7382	0.7001	**0.8080**	**0.7273**
Glove (2.2M vocabulary size)	ATAE-LSTM	0.6870$^{\diamond}$	NA	0.7720$^{\diamond}$	NA
	IAN	0.7210$^{\diamond}$	NA	0.7860$^{\diamond}$	NA
	DSAN-sum	0.7382	0.6961	**0.8009**	0.7109
	DSAN-mean	**0.7461**	**0.7052**	0.7964	**0.7136**

Table 3. Results of different methods on Laptop and Restaurant datasets with skip-gram word embeddings. Best results in each group are in bold.

Word embeddings	Model	Laptop		Restaurant	
		Accuracy	Macro-F1	Accuracy	Macro-F1
Skipgram	TD-LSTM	0.7179	0.6665	0.7848	0.6812
	MemNet	0.7257	0.6765	0.8027	0.7076
	RAM	0.7445	0.7009	0.7884	0.6835
	ATAE-LSTM	0.7257	0.6833	0.7866	0.6802
	IAN	0.7194	0.6664	0.7991	0.7046
	DSAN-sum	**0.7696**	**0.7265**	**0.8143**	**0.7311**
	DSAN-mean	0.7633	0.7136	0.8134	0.7235

if the DSAN model is calculated with single-layer attention, it also can be achieve better performance.

(2) In the second group of Table 2, the accuracies of the ATAE-LSTM, IAN, and DSAN models gradually increase, which demonstrate the importance of aspect information for aspect-level sentiment classification. All three models use single-layer attention mechanism, but they have shown an increasing trend in the use of aspect information. The ATAE-LSTM method simply uses the word embedding of the aspect as the aspect representations. IAN method utilizes LSTM to model the aspect themselves, and the DSAN model utilizes bidirectional GRU to model the aspect and the descriptive information of aspect. The experimental results of the ATAE-LSTM, IAN and

DSAN models show that the increase in the use of aspect information in the attention model helps the model to achieve better results.

(3) Word embeddings trained on domain-specific corpus are very helpful to the final classification results. After using the skipgram word embeddings, the MemNet, ATAE-LSTM and DSAN models all benefit from the domain knowledge, which have a great performance improvement on the Laptop and Restaurant datasets. The TD-LSTM and IAN models also have improvements on the Restaurant dataset to some extent, and performance is closed on the Laptop dataset. In this paper, the reproducible experimental results of the RAM model are not as good as those of the original author Chen et al. [2] on the published GloVe word vector, which may be related to the hyper-parameter setting of the neural network. Chen et al. [2] did not mention their hyperparameter settings about the RAM model.

(4) The results of the two merge modes of DSAN model are similar in the three sets of word embeddings. From the calculation of Formula 7, we can see that the difference between the *sum* mode and the *mean* mode is the scale factor m, which is the length of the aspect sub-sentence. Since the $softmax$ function computes each element in the attention weights in an exponential form, the local part of the words will be assigned more attentional weights in the sum model. It re-scales this concentration of the attentional weights, and the distribution of weights is relatively uniform in the mean mode. However, the dimensional factors have been scaled when calculating the attention scores of aspect sub-sentence and the sentence, and the length of most of the aspect sub-sentences are less than 5. So the classification of these two methods exhibit similar performance.

3.3 Effects of Dependency Subtree

In this subsection, we design a set of model comparison experiments to analyze the effect of the dependency subtree in the DSAN model. There are three different models in the comparison experiments. The aspect is represented by the word embedding in the first model, denoted as **W-AN**. The second model, denoted as **A-AN**, uses bidirectional GRU to model the aspect, and it doesn't include the descriptive information of the aspect. The third model is DSAN model. In the A-AN and DSAN model, the merge mode is *sum*, so we refer them as **A-AN-sum** and **DSAN-sum** for short.

From Table 4, the performance of W-AN, A-AN-sum, and DSAN-sum on the Laptop and Restaurant datasets increases sequentially, which reflects their increasing utilization degree of the aspect information. More accurate that the aspect representations are catched, more better that the performance of the attention model in the aspect-level sentiment classification task will achieve. The experimental results fully proves the importance of the dependency subtree of the aspect in the DSAN model.

Table 4. Effects of dependency subtree in DSAN model. Best results in each group are in bold.

Word embeddings	Model	Laptop		Restaurant	
		Accuracy	Macro-F1	Accuracy	Macro-F1
Skipgram	W-AN	0.7429	0.7061	0.7955	0.6992
	A-AN-sum	0.7571	0.7116	0.8054	0.7123
	DSAN-sum	**0.7696**	**0.7265**	**0.8143**	**0.7311**

3.4 Effects of Position Weight

As mentioned in Sect. 2.3, when different aspects are mentioned in a common sentence, the position information of the aspect is a very important feature that can help the model to distinguish different aspects from each other in the same sentence and make an improvement on the performance in the model.

Therefore, we introduce the position information of the aspect in the form of position weights, and define the distance of the aspect and the context word as the path length in dependency tree. We verify the effects of the position weights in two different word embeddings. Table 5 shows the experiment results of the effect of the position weights in different word embeddings. The experimental results prove that the position information of aspect can help model to better identify different aspects in the same sentence, thereby the model can achieve better performance. So, designing more effective ways of using position information is a worthwhile future work, such as position embedding.

Table 5. Effects of position weights in two different word embeddings. The postposition *base* means without position weights, and *position* means with position weights in the model. Best results in each group are in bold.

Word embeddings	Model	Laptop		Restaurant	
		Accuracy	Macro-F1	Accuracy	Macro-F1
Glove (2.2M vocabulary size)	DSAN-sum base	0.7288	0.6872	0.7982	0.7087
	DSAN-sum position	**0.7382**	**0.6961**	**0.8009**	**0.7109**
	DSAN-mean base	0.7351	0.6958	**0.7964**	0.7081
	DSAN-mean position	**0.7461**	**0.7052**	**0.7964**	**0.7136**
Skipgram	DSAN-sum base	0.7665	0.7213	0.8107	0.7209
	DSAN-sum position	**0.7696**	**0.7265**	**0.8143**	**0.7311**
	DSAN-mean base	0.7586	**0.7142**	0.8098	**0.7248**
	DSAN-mean position	**0.7633**	0.7136	**0.8134**	0.7235

3.5 Visualize Attention

In this subsection, we pick a review context *"Great food but the service was dreadful!"* from the Restaurant datasets as an example to visualize the attention weights of DSAN model on different aspect in the same sentence. There are two aspects: *"food"* and *"service"*, whose sentiment polarities are positive and negative respectively. We predict them by Skipgram word embeddings with the DSAN-sum and DSAN-mean model. Figures 3 and 4 show the attention weights of the DSAN-sum and DSAN-mean model on these two different aspects in the same sentence.

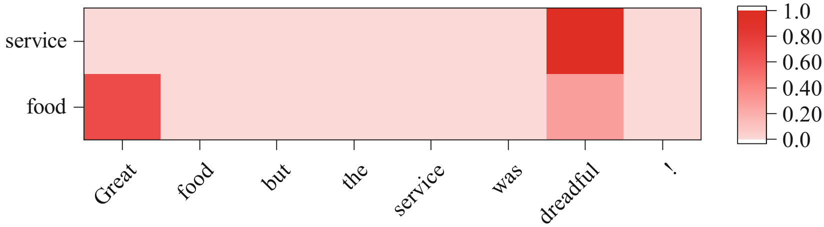

Fig. 3. Attention weights of DSAN-sum model on *"food"* and *"service"*.

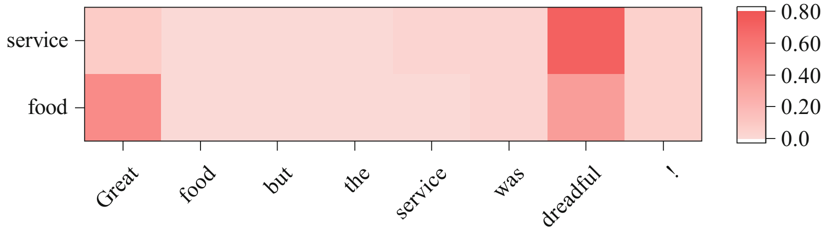

Fig. 4. Attention weights of DSAN-mean model on *"food"* and *"service"*.

From Figs. 3 and 4, we can clearly see the difference between the attention weights of two aspects. Both DSAN-sum and DSAN-mean show a high degree of attention to aspect-related sentiment expression, and assign most of the attentional weights, which helps to correctly predict the sentiment of these two aspects.

In Fig. 3, the DSAN-sum model assigns most of the attention weights to one or two words. When the DSAN-sum model predicts the aspect *"service"*, the word *"dreadful"* gets more than 90% of the attention weights, which helps the sentiment polarity of the aspect *"service"* to be correctly predicted to be negative. When the DSAN-sum model predicts the aspect *"food"*, the words *"Great"* and *"dreadful"* also get higher weights, but the word *"Great"* has much greater attention weights than the word *"dreadful"*, and the sentiment polarity of the aspect *"food"* is correctly predicted to be positive.

In Fig. 4, the DSAN-mean model assigns the attention weights more uniformly than DSAN-sum relatively. The DSAN-mean model also assigns most of the attention weights to the word *"dreadful"* for the aspect *"service"*, but it assigns a little weights to the word *"Great"*. When the DSAN-mean model predicts the aspect *"food"*, the word *"Great"* gets only a little more attention weights than the word *"dreadful"*. Although the DSAN-mean model correctly predicts the aspect *"food"*, the probability of the positive category is only slightly higher than the probability of the negative category.

From the visualization of the attention weights of the DSAN-sum and DSAN-mean model, the performance of the two models is the same as we analyzed in Sect. 3.2. The *sum* merge mode of the attention scores makes the model focus on a local part which gets the major attention weights. The *mean* merge mode makes the attention weights uniform relatively, because it already exists a scale factor $\sqrt{d_k}$ in the attention function, so the performance of two merge mode is similar.

4 Conclusions

In this paper, we propose a dependency subtree attention network (DSAN) model to determine the sentiment polarity of aspect in a sentence. On one hand, DSAN model can extract the dependency subtree that contains the descriptive information of the aspect based on the dependency tree of the sentence, which can offer more accurate aspect representations and assign e appropriate attention weights to the context words. On the other hand, DSAN model can better distinguish multiple aspect from each other in the same sentence by introducing the syntactic distance between aspect and context words. The experimental results on Laptop and Restaurant datasets show that the descriptive information of aspect can be more helpful to correctly predict aspect sentiment.

Acknowledgments. This paper is supported by the Applied Scientific and Technological Special Project of Department of Science and Technology of Guangdong Province (20168010124010); Natural Science Foundation of Guangdong Province (2015A030310318); Medical Scientific Research Foundation of Guangdong Province (A2015065).

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: International Conference on Learning Representations 2015 (2015)
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. 463–472 (2017)
3. Cheng, J., Zhao, S., Zhang, J., King, I., Zhang, X., Wang, H.: Aspect-level sentiment classification with heat (hierarchical attention) network. In: Proceedings of the 26th ACM International Conference on Information and Knowledge Management, pp. 97–106 (2017)

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. Gehring, J., Auli, M., Grangier, D., Yarats, D., Dauphin, Y.N.: Convolutional sequence to sequence learning. In: International Conference on Machine Learning 2017, pp. 1243–1252 (2017)
6. He, R., Mcauley, J.: Ups and downs: modeling the visual evolution of fashion trends with one-class collaborative filtering. In: International Conference on World Wide Web, pp. 507–517 (2016)
7. Jiang, L., Yu, M., Zhou, M., Liu, X., Zhao, T.: Target-dependent twitter sentiment classification. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pp. 151–160 (2011)
8. Kingma, D.P., Ba, J.L.: Adam: a method for stochastic optimization. In: International Conference on Learning Representations 2015 (2015)
9. Ma, D., Li, S., Zhang, X., Wang, H., Ma, D., Li, S., Zhang, X., Wang, H.: Interactive attention networks for aspect-level sentiment classification. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence, pp. 4068–4074 (2017)
10. Mikolov, T., Chen, K., Corrado, G.S., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
11. Nguyen, T.H., Shirai, K.: PhraseRNN: phrase recursive neural network for aspect-based sentiment analysis. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 2509–2514 (2015)
12. Pennington, J., Socher, R., Manning, C.D.: Glove: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, pp. 1532–1543 (2014)
13. Pontiki, M., Galanis, D., Pavlopoulos, J., Papageorgiou, H., Androutsopoulos, I., Manandhar, S.: Semeval-2014 task 4: aspect based sentiment analysis. In: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pp. 27–35 (2014)
14. Tang, D., Qin, B., Feng, X., Liu, T.: Effective lstms for target-dependent sentiment classification. In: International Conference on Computational Linguistics, pp. 3298–3307 (2016)
15. Tang, D., Qin, B., Liu, T.: Aspect level sentiment classification with deep memory network. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 214–224 (2016)
16. Vaswani, A., Shazeer, N., Parmar, N., Jones, L., Uszkoreit, J., Gomez, A.N., Kaiser, Ł.: Attention is all you need. In: Neural Information Processing Systems 2017, pp. 6000–6010 (2017)
17. Wang, Y., Huang, M., Zhu, X., Zhao, L.: 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)
18. Yin, Y., Song, Y., Zhang, M.: Document-level multi-aspect sentiment classification as machine comprehension. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 2044–2054 (2017)
19. Zhang, M., Zhang, Y., Vo, D.T.: Gated neural networks for targeted sentiment analysis. In: AAAI 2016 Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, pp. 3087–3093 (2016)