Aspect Sentiment Classification with both Word-level and Clause-level Attention Networks

Jingjing Wang[1], **Jie Li**[3], **Shoushan Li**[1,*], **Yangyang Kang**[2], **Min Zhang**[1], **Luo Si**[2], **Guodong Zhou**[1]
[1]School of Computer Science and Technology, Soochow University, China
[2]Alibaba Group, China
[3]School of Computer Science and Engineering, Southeast University, China
[1]djingwang@gmail.com, {lishoushan, minzhang, gdzhou}@suda.edu.cn,
[3]jennyetjieli@gmail.com, [2]{yangyang.kangyy, luo.si}@alibaba-inc.com

Abstract

Aspect sentiment classification, a challenging task in sentiment analysis, has been attracting more and more attention in recent years. In this paper, we highlight the need for incorporating the importance degrees of both words and clauses inside a sentence and propose a hierarchical network with both word-level and clause-level attentions to aspect sentiment classification. Specifically, we first adopt *sentence-level discourse segmentation* to segment a sentence into several clauses. Then, we leverage multiple Bi-directional LSTM layers to encode all clauses and propose a word-level attention layer to capture the importance degrees of words in each clause. Third and finally, we leverage another Bi-directional LSTM layer to encode the output from the former layers and propose a clause-level attention layer to capture the importance degrees of all the clauses inside a sentence. Experimental results on the *laptop* and *restaurant* datasets from SemEval-2015 demonstrate the effectiveness of our proposed approach to aspect sentiment classification.

1 Introduction

The past decade has witnessed an exploding interest in sentiment analysis from natural language processing and data mining communities due to its inherent challenges and wide applications [Pang and Lee, 2007; Liu, 2012]. As a fine-grained sentiment classification task, aspect sentiment classification aims to identify the sentiment polarity for a particular aspect. For example, the sentence "*The price was too high but the food was delicious*" would be assigned with *negative* polarity for aspect "*price*" while with *positive* polarity for aspect "*food*". Early studies typically employ traditional supervised learning algorithms which focus on designing a bag of features such as bag-of-words to train a classifier (e.g., Support Vector Machine, SVM) [Jiang *et al.*, 2011; Pérez-Rosas *et al.*, 2012].

*Corresponding author

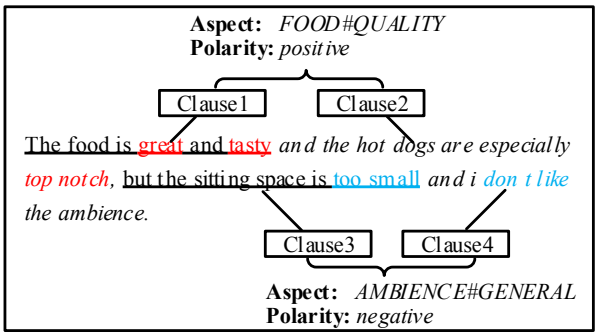


Figure 1: An example sentence in the *restaurant* domain where the entity *E* and attribute *A* pair (i.e., *E#A*) defines the aspect category of the given text.

Recently, neural network approaches have shown promising results on sentiment classification, such as Recursive NN [Socher *et al.*, 2011], Recursive NTN [Socher *et al.*, 2013] and Tree-LSTM [Tai *et al.*, 2015]. However, above neural network based approaches for sentiment classification just make use of the contexts without consideration of the aspect information while the aspect information should be an important factor for judging the aspect sentiment polarity. One possible way to incorporate the aspect information is to distinguish the importance of different texts with respect to a specific aspect.

First, for a specific aspect, the importance degrees of different words are different. For instance, in Figure 1, the words such as "*great*", "*tasty*" contribute much in implying the *positive* sentiment polarity for the aspect *FOOD#QUALITY*. While, the words such as "*is*", "*and*" don't contribute. Therefore, a well-behaved neural network approach should consider the importance degrees of different words for predicting the sentiment polarity of a specific aspect.

Second, for a particular aspect, the importance degrees of different clauses are different. For instance, in Figure 1, the first and second clauses have much stronger information in assisting the prediction of the sentiment polarity for the aspect *FOOD#QUALITY*. In contrast, the third and fourth clauses are more relevant to the aspect *AMBIENCE#GENERAL*. Therefore, a well-behaved neural net-

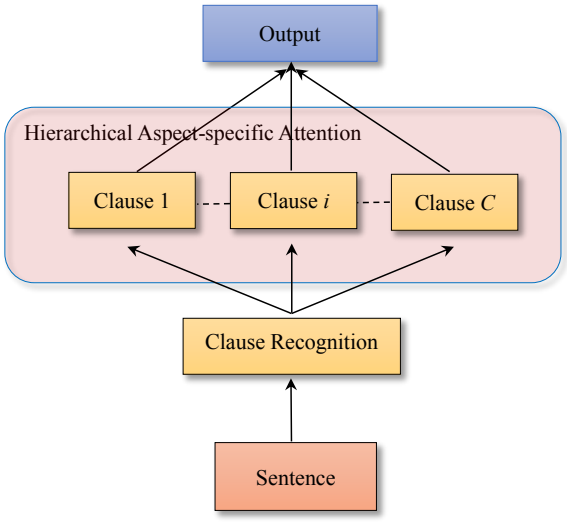


Figure 2: The overview of our approach

work approach should consider the importance degrees of different clauses for predicting the sentiment polarity of a specific aspect.

In particular, we propose a neural architecture, i.e., the Hierarchical Aspect-specific Attention Network, which leverages both word-level and clause-level attentions to incorporate the importance degrees of both words and clauses in a sentence. First, we adopt a *sentence-level discourse segmentation* method to segment a sentence into several clauses. Then, we leverage multiple Bi-directional LSTM layers to encode all clauses and propose a word-level attention layer to capture the importance degrees of words in each clause. Third, we leverage another Bi-directional LSTM to encode the output from the former Bi-directional LSTM layers and propose a clause-level attention layer to capture the importance degrees of all clauses. Experimental results on the *laptop* and *restaurant* datasets from SemEval-2015 [Pontiki *et al.*, 2015] demonstrate that our proposed approach outperforms a number of competitive baselines and even significantly performs better than the best-performed system *Sentiue* in the shared task of SemEval-2015 [Saias, 2015].

2 Related Work

2.1 Aspect Sentiment Classification

In the literature, aspect sentiment classification is typically regarded as a text classification problem. Therefore, text classification approaches, such as SVM [Jiang *et al.*, 2011], can be naturally applied to solve the aspect sentiment classification task without consideration of the mentioned aspect. Traditional machine learning approaches mainly focus on feature engineering to train a sentiment classifier [Jiang *et al.*, 2011; Pérez-Rosas *et al.*, 2012] and unable to discover the discriminative or explanatory factors of data. To solve this problem, [Dong *et al.*, 2014] transfer a dependency tree of a sentence into a target-specific recursive structure, and get higher level representation based on that structure. [Vo and Zhang, 2015] use rich features including sentiment-specific word embedding and sentiment lexicons. [Guan *et al.*, 2016] propose a novel deep learning framework for review sentiment classification which employs prevalently available ratings as weak supervision signals. [Tang *et al.*, 2016b] propose a neural based approach that determines sentiment towards a target word based on its position.

2.2 Aspect Sentiment Classification with Neural Networks

Recently, neural network approaches have shown promising results on sentiment classification, such as Recursive NN [Socher *et al.*, 2011], Recursive NTN [Socher *et al.*, 2013] and Tree-LSTM [Tai *et al.*, 2015]. However, the neural network based approaches just make use of the contexts without consideration of aspects which also make great contributions to judging the sentiment polarity of aspect.

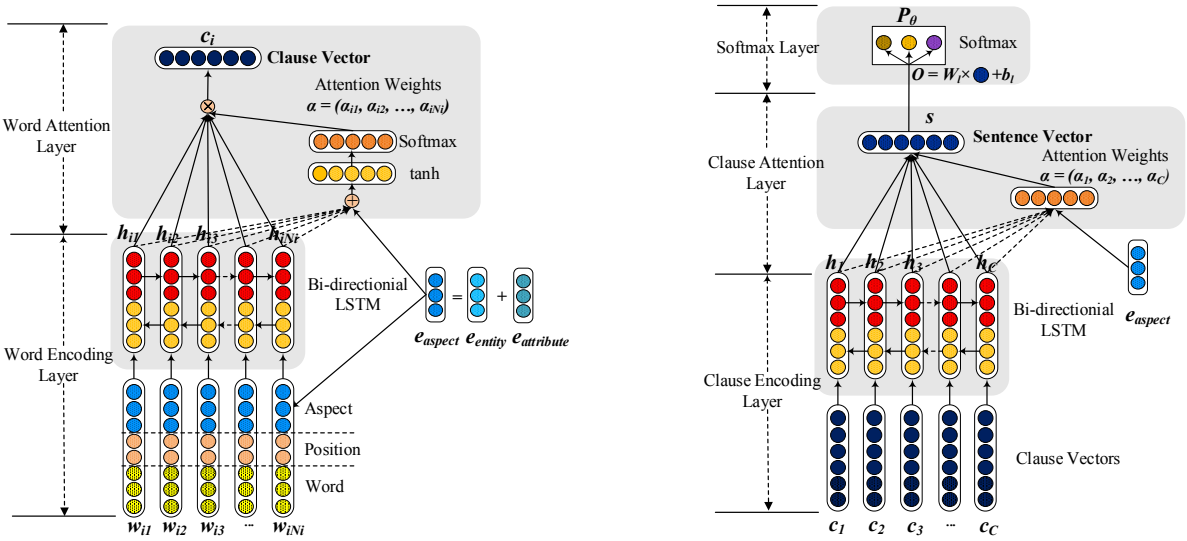
Therefore, in order to incorporate aspects into a model, [Tang *et al.*, 2016a] develop two long short-term memory (LSTM) to model the left and right contexts with target. [Wang *et al.*, 2016] propose an attention-based LSTM in order to explore the potential correlation of aspects and sentiment polarities in aspect sentiment classification. [Tang *et al.*, 2016b] design deep memory networks which consist of multiple computational layers to integrate the target information. [Chen *et al.*, 2017] also propose a deep memory network to integrate the target information, but the results of multiple attentions are non-linearly combined with a recurrent neural network. [Ma *et al.*, 2017] propose an interactive learning approach, which interactively learns attentions in the contexts and targets.

Although above deep neural network models have achieved great success on aspect sentiment classification, they all ignore the incorporating knowledge of clause-level information in the model architecture. To the best of our knowledge, we are the first to address aspect sentiment classification with both word-level and clause-level attentions.

3 Hierarchical Aspect-specific Attention Network

In this section, we first introduce a clause recognition method to segment a sentence into several clauses. Then, we propose a hierarchical aspect-specific attention model which can concentrate on both the informative words and clauses corresponding to a given aspect in detail. Figure 2 shows the overview of the proposed approach to aspect sentiment classification.

Clause recognition is a non-trivial problem. Fortunately, in the literature, the clause recognition could be seen as a sub-problem of *discourse segmentation* which has been well-studied in the NLP community. Specifically, *discourse segmentation* is the task of breaking a given text into non-overlapping segments called elementary discourse units (EDUs) [Carlson *et al.*, 2001]. Each EDU could be seen as a clause. In this study, we employ *sentence-level discourse segmentation*, which aims to segment a sentence into EDUs [Soricut and Marcu, 2003]. There exist several kinds of discourse theories and each of them has its own specificities in terms of segmentation guidelines and size of units. In this study, we adopt Rhetorical Structure Theory (RST) [MANN,



(a) Word-level Aspect-specific Attention Module

(b) Clause-level Aspect-specific Attention Module

Figure 3: The overall architecture of our proposed hierarchical aspect-specific attention approach

1988] due to its well-defined EDUs and perform sentence-level discourse segmentation to detect EDUs as clauses. For instance, after the sentence-level discourse segmentation, the example in Figure 1 is segmented into four non-overlapping clauses, i.e., $1A$, $1B$, $1C$ and $1D$, as shown in **E1**.

E1: [The food is great and tasty]^{1A} [and the hot dogs are especially top notch,]^{1B} [but the setting space is too small]^{1C} [and i don't like the ambience.]^{1D}

In the following, we introduce our hierarchical aspect-specific attention model to extract both the informative words and clauses corresponding to the specific aspect. Figure 3 shows the overall architecture of this approach which mainly consists of two components, i.e., a word-level aspect-specific attention module and a clause-level aspect-specific attention module. We will describe the details of the two modules as follows.

3.1 Word-level Aspect-specific Attention

Word Encoding Layer. Assume that a sentence has been segmented into C clauses c_i and each clause contains N_i words. I_{ij} represents the j-th word in the i-th clause. Given a clause c_i with the word I_{ij}, the vector representation $w_{ij} \in \mathbb{R}^{d=d_w+d_p}$ of word I_{ij} consists of its word embedding and position embedding [Zeng *et al.*, 2014], which is calculated as $w_{ij} = E_w \cdot I_{ij} \oplus E_p \cdot I_{ij}$ where $E_w \in \mathbb{R}^{d_w \times |V|}$ is word embedding matrix and $E_p \in \mathbb{R}^{d_p \times |V'|}$ is position embedding matrix.

An aspect category consists of an entity and an attribute [Pontiki *et al.*, 2015]. Specifically, the entity string e_{entity} of length L_1 is represented as $\{x_1, ..., x_{L_1}\}$ where $x_n \in \mathbb{R}^{d'}$ is the d'-dimensional vector of the n-th word in the entity string. The attribute string $e_{attribute}$ is represented as $\{z_1, ..., z_{L_2}\}$. Since the common word embedding representations exhibit linear structure that makes it possible to meaningfully combine words by an element-wise addition of their vector representations, we use the sum of the entity and attribute embeddings to obtain a more compact aspect representation, i.e.,

$$e_{aspect} = e_{entity} + e_{attribute} = \frac{1}{L_1}\sum_{n=1}^{L_1} x_n + \frac{1}{L_2}\sum_{n=1}^{L_2} z_n \quad (1)$$

Then, inspired by [Tang *et al.*, 2016a], we append the aspect representation to the embedding of each word to form an aspect-augmented embedding for each word j, i.e.,

$$\hat{w}_{ij} = w_{ij} \oplus e_{aspect}; \quad \hat{w}_{ij} \in \mathbb{R}^{d+d'},\ i \in [1, C],\ j \in [1, N_i] \quad (2)$$

where $\oplus$ denotes the concatenate operator. C is the number of clauses and N_i is the number of words in the clause c_i. Noted that the dimension of $\hat{w}_{ij}$ is $(d + d')$.

Then, we use a Bi-directional LSTM (namely, Bi-LSTM) [Graves *et al.*, 2013], which can efficiently make use of past features (via forward states) and future features (via backward states) for a specific time frame, to get annotations of words by summarizing information from both directions for words. The Bi-LSTM contains the forward LSTM $\overrightarrow{f}$ which reads the clause c_i from the word $I_{i,1}$ to I_{i,N_i} and a backward LSTM $\overleftarrow{f}$ which reads from I_{i,N_i} to $I_{i,1}$:

$$\overrightarrow{h}_{ij} = \overrightarrow{\text{LSTM}}(\hat{w}_{ij}); \quad i \in [1, C],\ j \in [1, N_i] \quad (3)$$

$$\overleftarrow{h}_{ij} = \overleftarrow{\text{LSTM}}(\hat{w}_{ij}); \quad i \in [1, C],\ j \in [N_i, 1] \quad (4)$$

We obtain an annotation for a given word I_{ij} by concatenating the forward hidden state $\overrightarrow{h}_{ij}$ and backward hidden state $\overleftarrow{h}_{ij}$ as follows:

$$h_{ij} = \overrightarrow{h}_{ij} \oplus \overleftarrow{h}_{ij} \quad (5)$$

which summarizes the information of the whole clause centered around the word I_{ij}.

Word Attention Layer. Traditional LSTM model cannot capture the information about which words are important to the meaning of the clause. In order to address this problem, we design an attention mechanism which drives the model to concentrate on such words in the clause with respect to a specific aspect and aggregate the representation of those informative words to form a clause vector.

Figure 3(a) shows the details of the word-level attention module. Specifically, the following formulas are applied to compute the attention weight α_{ij} (similarity or relatedness) between each word annotation h_{ij} and the aspect representation e_{aspect}.

$$u_{ij} = \tanh(W_w \cdot [h_{ij};\ e_{aspect}] + b_w) \tag{6}$$

$$\alpha_{ij} = \mathrm{softmax}(u_{ij}) = \frac{\exp(u_{ij})}{\sum_{t=1}^{N} \exp(u_{it})} \tag{7}$$

where $[h_{ij};\ e_{aspect}]$ denotes the vertical concatenation of h_{ij} and e_{aspect}, $1 \le j \le N_i$, W_w is an intermediate matrix and b_w is an offset value. $\alpha = [\alpha_{i1}, \alpha_{i2}, ..., \alpha_{iN_i}]$ are the weight vector of all words.

Then we compute the clause vector c_i as a weighted sum of the word annotations based on the weights, i.e.,

$$c_i = \sum_{i=1}^{N_i} \alpha_{ij} \cdot h_{ij} \tag{8}$$

3.2 Clause-level Aspect-specific Attention

Clause Encoding Layer. Given the clause vectors c_i, we also use a Bi-LSTM to encode the clauses in order to incorporate the contextual information in the annotations, i.e.,

$$\overrightarrow{h}_i = \overrightarrow{\mathrm{LSTM}}(c_i); \quad i \in [1, C] \tag{9}$$

$$\overleftarrow{h}_i = \overleftarrow{\mathrm{LSTM}}(c_i); \quad i \in [C, 1] \tag{10}$$

Similarly, we obtain an annotation for the clause c_i by concatenating $\overrightarrow{h}_i$ and $\overleftarrow{h}_i$ as follows:

$$h_i = \overrightarrow{h}_i \oplus \overleftarrow{h}_i \tag{11}$$

which summarizes the information of the whole sentence centered around the clause c_i.

Clause Attention Layer. Figure 3(b) shows the details of the clause-level attention module. In this figure, $[h_1, h_2, ..., h_C]$ are annotation vectors for the clauses. With these context clause representations, we compute the attention weight α_i between each clause annotation h_i and the aspect representation e_{aspect} as follows:

$$m_i = \tanh(W_c \cdot [h_i;\ e_{aspect}] + b_c) \tag{12}$$

$$\alpha_i = \mathrm{softmax}(m_i) = \frac{\exp(m_i)}{\sum_{t=1}^{C} \exp(m_t)} \tag{13}$$

where $1 \le i \le C$, W_c is an intermediate matrix, b_c is an offset value. In addition, e_{aspect} is the same as that in Equation (6), which is also calculated according to Equation (1).

After computing the clause annotation weights, we can get the sentence representation s based on the attention vectors α_i by:

$$s = \sum_{i=1}^{C} \alpha_i \cdot h_i \tag{14}$$

Softmax Layer. To perform aspect sentiment classification, we feed the sentence representation s to a softmax classifier, i.e.,

$$o = W_l \cdot s + b_l \tag{15}$$

where $o \in \mathbb{R}^K$ is the output, W_l is the weight matrix and b_l is the bias. Then, the probability of labeling sentence with sentiment polarity $k \in [1, K]$ is computed by:

$$p_\theta = \frac{\exp(o_k)}{\sum_{t=1}^{K} \exp(o_t)} \tag{16}$$

where θ denotes all parameters. Finally, the label with the highest probability stands for the predicted sentiment polarity of the aspect.

3.3 Model Training

We use cross-entropy loss function to train our model end-to-end given a set of training data x_t, e_t, y_t, where x_t is the t-th text to be predicted, e_t is the corresponding aspect and y_t is one-hot representation of the ground-truth sentiment polarity for aspect e_t and text x_t. We represent this model as a black-box function $\phi(x, e)$ whose output is a vector representing the probability of sentiment polarity. The goal of training is to minimize the loss function:

$$J(\theta) = -\sum_{t=1}^{M} \sum_{k=1}^{K} y_t^k \cdot \log \phi(x_t, e_t) + \frac{l}{2} ||\theta||_2^2 \tag{17}$$

where M is the number of training samples; K is the category number and l is a L_2 regularization to bias parameters.

In the above equation, the model parameters are optimized by using Adagrad [Duchi *et al.*, 2011]. All the matrix and vector parameters are initialized with uniform distribution in $[-\sqrt{6/(r+c')}, \sqrt{6/(r+c')}]$, where r and c' are the numbers of rows and columns in the matrices [Glorot and Bengio, 2010]. The dropout strategy is used in the Bi-directional LSTM layer in order to avoid over-fitting.

4 Experimentation

4.1 Experimental Settings

• **Data Settings:** We conduct experiments on two datasets (i.e., one from the *laptop* domain and the other from the *restaurant* domain) from SemEval-2015 Task 12[1] [Pontiki *et al.*, 2015] to validate the effectiveness of our approach. Each dataset consists of many customers reviews and each review contains a list of aspects and corresponding sentiment polarities, i.e., *positive*, *neutral* or *negative*. We also set aside 10% from the training set as the development data which is used to tune algorithm parameters.

[1]The detail introduction of this task is available at http://alt.qcri.org/semeval2015/task12/

Model	*Restaurant* (*Accuracy*)	*Laptop* (*Accuracy*)
Majority	0.537	0.570
LSTM [Wang *et al.*, 2016]	0.735	0.734
TC-LSTM [Tang *et al.*, 2016a]	0.747	0.745
ATAE-LSTM [Wang *et al.*, 2016]	0.752	0.747
RAM [Chen *et al.*, 2017]	0.767	0.759
IAN [Ma *et al.*, 2017]	0.755	0.753
Sentiue [Saias, 2015]	0.787	0.793
Hierarchical Bi-LSTM	0.763	0.767
Word-Level ATT	0.789	0.785
Clause-Level ATT	0.783	0.779
Word&Clause-Level ATT	**0.809**	**0.816**

Table 1: *Accuracy* on aspect sentiment classification about both the *restaurant* and *laptop* domains

Model	*Restaurant* (*Macro-F1*)	*Laptop* (*Macro-F1*)
Majority	0.233	0.242
LSTM [Wang *et al.*, 2016]	0.617	0.608
TC-LSTM [Tang *et al.*, 2016a]	0.634	0.622
ATAE-LSTM [Wang *et al.*, 2016]	0.641	0.637
RAM [Chen *et al.*, 2017]	0.645	0.639
IAN [Ma *et al.*, 2017]	0.639	0.625
Sentiue [Saias, 2015]	0.660	0.634
Hierarchical Bi-LSTM	0.647	0.632
Word-Level ATT	0.662	0.646
Clause-Level ATT	0.659	0.647
Word&Clause-Level ATT	**0.685**	**0.667**

Table 2: *Macro-F1* on aspect sentiment classification about both the *restaurant* and *laptop* domains

• **Word Representations: (1) PTE:** This is a word embedding resource built by ourselves with PTE which is a semi-supervised representation learning tool proposed by [Tang *et al.*, 2015][2]. This tool could leverage both labeled and unlabeled data to build a large-scale heterogeneous network and use the network to train the word vectors. In our implementation, on one hand, the labeled data is collected from Amazon by [McAuley *et al.*, 2015]. Specifically, we pick the domains, i.e., *Books*, *CDs*, *Clothing*, *Electronics*, *Restaurant* and *Health* and each review is automatically assigned with a *positive* category if its rating score is 4 or 5 and a *negative* category if its rating score is 1 or 2. On the other hand, the unlabeled data is the data from SemEval-2015 Task, as introduced above. The vocabulary size is about 1.2 million and the dimensionality of word vector is 300. **(2) Position Embeddings:** Inspired by [Zeng *et al.*, 2014], we use position embeddings to specify the aspect words, i.e., entity and attribute words (if available, in the sentence). The position embedding corresponds to the relative distance from current word to the aspect word. For instance, in Figure 1, the relative distance from the word "*great*" to the aspect word "*food*" is 3. In our experiments, the relative distance is mapped to a vector of dimension 100.

• **EDU:** We run EDU splitting with the Discourse Segmenter Tool[3] on all the datasets.

• **Hyper-parameters:** In our experiments, the word embedding and position embedding are optimized during training. All out-of-vocabulary words are initialized by sampling from the uniform distribution $U(-0.01, 0.01)$. The dimensions of attention vectors and LSTM hidden states are set to be 300. The other hyper-parameters are tuned according to the development data. Specifically, the initial learning rate is 0.1. The regularization weight of the parameters is 10^{-5}, and the dropout rate is set to 0.25.

• **Evaluation Metrics:** The performance is evaluated using *Accuracy* and *Macro-F1* (F) which is calculated as $F = \frac{2PR}{P+R}$, where the overall precision P and recall R are averaged on the precision/recall scores of all categories.

[2]The word embedding resource is released at https://github.com/jjwangnlp/PTE2ASC

[3]http://alt.qcri.org/tools/discourse-parser/

4.2 Experimental Results

In this subsection, we give some baseline approaches for comparison in order to comprehensively evaluate the performance of our proposed approach. Note that all these learning approaches employ the same representations, i.e., word PTE embedding together with the position embedding.

• **Majority:** This approach is a basic baseline approach, which assigns the majority sentiment polarity in the training set to each sample in the test set.

• **LSTM:** This approach only uses one LSTM network to model the context. After that, the average value of all the hidden states is fed to a softmax function to estimate the probability of each sentiment label [Wang *et al.*, 2016].

• **TC-LSTM:** This approach extends LSTM by taking into account of the aspect information where two LSTM networks, a forward one and backward one towards the aspect, are adopted. This is a state-of-the-art approach to aspect sentiment classification proposed by [Tang *et al.*, 2016a].

• **ATAE-LSTM:** This approach models the context words via attention-based LSTM and appends the aspect embeddings with each word embedding vector. This is a state-of-the-art approach proposed by [Wang *et al.*, 2016].

• **RAM:** This approach captures importance of context words for a specific aspect with a deep memory network and the results of multiple attentions are non-linearly combined with a recurrent neural network. This is a state-of-the-art approach proposed by [Chen *et al.*, 2017].

• **IAN:** This approach is an interactive learning approach, which firstly models the contexts and aspects via LSTM and then interactively learns attentions in the contexts and aspects. This is another state-of-the-art approach proposed by [Ma *et al.*, 2017].

• **Sentiue:** This is the best-performed system in SemEval-2015 Task 12 [Saias, 2015]. It employs a MaxEnt classifier with n-gram features, POS features, lexicon features and achieves the best accuracy scores in both the domains *laptop* and *restaurant*.

• **Hierarchical Bi-LSTM:** Our approach which employs neither the word-level nor clause-level attention.

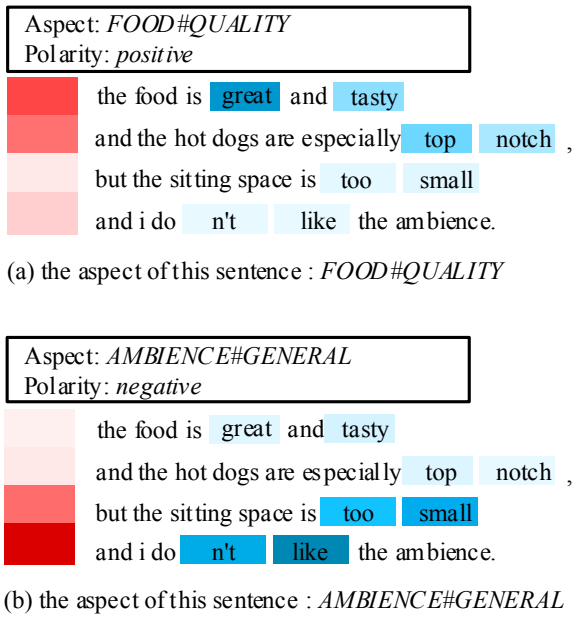


Figure 4: The attention visualizations for a sentence including two different aspects, i.e., "FOOD#QUALITY" and "AMBIENCE#GENERAL"

- **Word-Level ATT:** Our approach which employs only the word-level attention.
- **Clause-Level ATT:** Our approach which employs only the clause-level attention.
- **Word&Clause-Level ATT:** Our approach which employs both the word-level and clause-level attentions.

Table 1 and Table 2 show the performance comparison of different approaches. From these two tables, we can see that, all LSTM-based models perform better than the **Majority** approach, showing that LSTM has potentials in automatically generating representations and can all bring performance improvement for sentiment classification.

The four state-of-the-art approaches, i.e., **TC-LSTM**, **ATAE-LSTM**, **RAM** and **IAN**, all perform better than **LSTM**. These results confirm the helpfulness of considering aspect information in aspect sentiment classification. Moreover, we find that **ATAE-LSTM**, **RAM** and **IAN** perform a bit better than **TC-LSTM**, which demonstrates that using attention mechanism is a good choice to model aspect information.

Our approach **Hierarchical Bi-LSTM** outperforms most of the state-of-the-art approaches, which highlights the importance of employing clause information in a sentence. When the word-level or clause-level attention is leveraged, our approach **Word-Level ATT** and **Clause-Level ATT** achieve better performance and outperform all the state-of-the-art approaches. Among all these approaches, our approach **Word&Clause-Level ATT** performs best and even outperforms the top-performed system of SemEval-2015 Task 12, i.e., **Sentiue**. Impressively, compared to **LSTM**, **Word&Clause-Level ATT** achieves the average improvement of 7.4% (*Accuracy*), 6.8% (*Macro-F1*) on the *restaurant* dataset and 8.2% (*Accuracy*), 5.9% (*Macro-F1*) on the *laptop* dataset. These results encourage to incorporate the importance degrees of both words and clauses in a sentence.

4.3 Discussion: Visualization of Attention

In order to get a better understanding of our hierarchical aspect-specific attention model and validate that this model is able to select informative words and clauses corresponding to a specific aspect in a sentence, we visualize the word-level attention layers and clause-level attention layers according to the obtained attention weight α in Equation (7) and Equation (13) respectively.

Figure 4 shows the attention visualizations for a sentence including two different aspects, i.e., *FOOD#QUALITY* and *AMBIENCE#GENERAL*, from the *restaurant* dataset. Here, we use the visualization approach proposed by [Yang *et al.*, 2016]. Specifically, we normalize the word weight by the clause weight to make sure that only informative words in informative clauses corresponding to a given specific aspect are emphasized. In Figure 4, each line is a clause. Red denotes the clause weight and blue denotes the word weight. The color depth indicates the importance degree of attention weight for a specific aspect, the darker the more important.

From this figure, we can see that the clause-level attention function always selects the informative clauses corresponding to a specific aspect, such as selecting the first and second clauses for the aspect *FOOD#QUALITY*; while selecting the third and fourth clauses for the aspect *AMBIENCE#GENERAL*. In addition, the word-level attention function can select both the words and multi-word phrases carrying strong sentiment signals corresponding to a given specific aspect, such as *"great"*, *"tasty"*, *"top notch"* if the given aspect is *FOOD#QUALITY*; while *"too small"* and *"n't like"* if the given aspect is *AMBIENCE#GENERAL*.

5 Conclusion

In this paper, we propose a hierarchical aspect-specific attention model for aspect sentiment classification. The main idea of the proposed model is to segment a sentence into several clauses and then use both word-level and clause-level attentions to incorporate the knowledge of word-level and clause-level text information. Experimental results on the *laptop* and *restaurant* datasets from SemEval-2015 demonstrate that the proposed approach outperforms a number of competitive baselines and even the best-performed system in the shared task of SemEval-2015.

In our future work, we would like to employ more information in the discourse analysis, e.g., relationships between two clauses, to improve the performance. Furthermore, we would like to apply our word-level and clause-level model to other sentiment analysis tasks, e.g., sentence-level sentiment classification.

Acknowledgments

The research work is partially supported by the National Key R&D Program of China under Grant No.2017YFB1002101 and two NSFC grants No.61331011, No.61672366. This work is also supported by the joint research project of Alibaba and Soochow University.

References

[Carlson *et al.*, 2001] Lynn Carlson, Daniel Marcu, and Mary Ellen Okurovsky. Building a discourse-tagged corpus in the framework of rhetorical structure theory. In *Proceedings of SIGDIAL-2001*, 2001.

[Chen *et al.*, 2017] Peng Chen, Zhongqian Sun, Lidong Bing, and Wei Yang. Recurrent attention network on memory for aspect sentiment analysis. In *Proceedings of EMNLP-2017*, pages 452–461, 2017.

[Dong *et al.*, 2014] Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. Adaptive recursive neural network for target-dependent twitter sentiment classification. In *Proceedings of ACL-2014*, pages 49–54, 2014.

[Duchi *et al.*, 2011] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

[Glorot and Bengio, 2010] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of AISTATS-2010*, pages 249–256, 2010.

[Graves *et al.*, 2013] Alex Graves, Abdel Rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. 38(2003):6645–6649, 2013.

[Guan *et al.*, 2016] Ziyu Guan, Long Chen, Wei Zhao, Shulong Tan, Shulong Tan, and Deng Cai. Weakly-supervised deep learning for customer review sentiment classification. In *Proceedings of IJCAI-2016*, pages 3719–3725, 2016.

[Jiang *et al.*, 2011] Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. Target-dependent twitter sentiment classification. In *Proceedings of ACL-2011*, pages 151–160, 2011.

[Liu, 2012] Bing Liu. *Sentiment Analysis and Opinion Mining*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2012.

[Ma *et al.*, 2017] Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. Interactive attention networks for aspect-level sentiment classification. In *Proceedings of IJCAI-2017*, pages 4068–4074, 2017.

[MANN, 1988] W. MANN. Rhetorical structure theory : Toward a functional theory of text organization. *Text*, 8(3):243–281, 1988.

[McAuley *et al.*, 2015] Julian J. McAuley, Rahul Pandey, and Jure Leskovec. Inferring networks of substitutable and complementary products. In *Proceedings of SIGKDD-2015*, pages 785–794, 2015.

[Pang and Lee, 2007] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135, 2007.

[Pérez-Rosas *et al.*, 2012] Verónica Pérez-Rosas, Carmen Banea, and Rada Mihalcea. Learning sentiment lexicons in spanish. In *Proceedings of LREC-2012*, pages 3077–3081, 2012.

[Pontiki *et al.*, 2015] Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. Semeval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of NAACL-HLT-2015*, pages 486–495, 2015.

[Saias, 2015] José Saias. Sentiue: Target and aspect based sentiment analysis in semeval-2015 task 12. In *Proceedings of NAACL-HLT-2015*, pages 767–771, 2015.

[Socher *et al.*, 2011] Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, and Christopher D. Manning. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of EMNLP-2011*, pages 151–161, 2011.

[Socher *et al.*, 2013] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. 2013.

[Soricut and Marcu, 2003] Radu Soricut and Daniel Marcu. Sentence level discourse parsing using syntactic and lexical information. In *Proceedings of NAACL-2003*, 2003.

[Tai *et al.*, 2015] Kai Sheng Tai, Richard Socher, and Christopher D. Manning. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of ACL-2015*, pages 1556–1566, 2015.

[Tang *et al.*, 2015] Jian Tang, Meng Qu, and Qiaozhu Mei. PTE: predictive text embedding through large-scale heterogeneous text networks. In *Proceedings of SIGKDD-2015*, pages 1165–1174, 2015.

[Tang *et al.*, 2016a] Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. Effective lstms for target-dependent sentiment classification. In *Proceedings of COLING-2016*, pages 3298–3307, 2016a.

[Tang *et al.*, 2016b] Duyu Tang, Bing Qin, and Ting Liu. Aspect level sentiment classification with deep memory network. In *Proceedings of EMNLP-2016*, pages 214–224, 2016b.

[Vo and Zhang, 2015] Duy-Tin Vo and Yue Zhang. Target-dependent twitter sentiment classification with rich automatic features. In *Proceedings of IJCAI-2015*, pages 1347–1353, 2015.

[Wang *et al.*, 2016] Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. Attention-based LSTM for aspect-level sentiment classification. In *Proceedings of EMNLP-2016*, pages 606–615, 2016.

[Yang *et al.*, 2016] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. Hierarchical attention networks for document classification. In *Proceedings of NAACL-HLT-2016*, pages 1480–1489, 2016.

[Zeng *et al.*, 2014] Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. Relation classification via convolutional deep neural network. In *Proceedings of COLING-2014*, pages 2335–2344, 2014.