

Article

A Bi-Directional LSTM-CNN Model with Attention for Aspect-Level Text Classification

Yonghua Zhu ¹, Xun Gao ², Weilin Zhang ², Shenkai Liu ¹ and Yuanyuan Zhang ^{3,*}

¹ Shanghai Film Academy, Shanghai University, Shanghai 200444, China; zyh@shu.edu.cn (Y.Z.); kingsen@shu.edu.cn (S.L.)

² School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China; gx00727@shu.edu.cn (X.G.); zeroized@i.shu.edu.cn (W.Z.)

³ College of Information Technology, Zhejiang Chinese Medical University, Hangzhou 310053, China

* Correspondence: zyy@zjtcn.net; Tel.: +86-152-2161-8589

Received: 25 October 2018; Accepted: 22 November 2018; Published: 24 November 2018



Abstract: The prevalence that people share their opinions on the products and services in their daily lives on the Internet has generated a large quantity of comment data, which contain great business value. As for comment sentences, they often contain several comment aspects and the sentiment on these aspects are different, which makes it meaningless to give an overall sentiment polarity of the sentence. In this paper, we introduce Attention-based Aspect-level Recurrent Convolutional Neural Network (AARCNN) to analyze the remarks at aspect-level. The model integrates attention mechanism and target information analysis, which enables the model to concentrate on the important parts of the sentence and to make full use of the target information. The model uses bidirectional LSTM (Bi-LSTM) to build the memory of the sentence, and then CNN is applied to extracting attention from memory to get the attentive sentence representation. The model uses aspect embedding to analyze the target information of the representation and finally the model outputs the sentiment polarity through a softmax layer. The model was tested on multi-language datasets, and demonstrated that it has better performance than conventional deep learning methods.

Keywords: attention mechanism; NLP; aspect-level sentiment classification

1. Introduction

Aspect-level sentiment classification, or aspect-based sentiment analysis (ABSA), is a fine-grained sentence analysis, which recognizes the aspects mentioned in the sentence and their corresponding sentiment polarity [1]. ABSA task consists of two subtasks, aspect category detection and sentiment polarity classification. For instance, given the sentence “The food is not cheap but quite delicious”, first the two aspects of sentiment target food are extracted: FOOD#PRICE and FOOD#TASTE. Then, the sentiment for each aspect is classified: negative for “not cheap” in FOOD#PRICE and positive for “quite delicious” in FOOD#TASTE.

ABSA task has become one of the research focuses in the sentiment analysis domain of natural language processing (NLP) [2,3]. Conventional machine learning methods such as Support Vector Machine (SVM) have been applied to the text classification issues [4], and can also be used in sentiment analysis. These methods extract features in the sentence to build the model, and use these features to predict the categories of non-annotated data. There have been many achievements in the machine learning approaches [4,5]. However, conventional machine learning approaches have an obvious drawback that they treat the sentence as a bag of words and do not consider the sequence and the relationships between aspect terms and its corresponding opinion-indicating words, when the sentiment polarity is often determined by such sequence or relationships. Another problem is that these

models depend too much on the feature engineering and require a quantity of manual preprocessing, which stops them from being more efficient or accurate.

Jiang et al. [6] pointed out that considering target information tends to reduce errors about 40% in the sentiment classification tasks. Deep learning methods, especially recurrent neural network (RNN) [7], consider the context when modeling the sentence, which enable them to analyze target information and its context word dependencies. Dong et al. [8] introduced a self-adaptive recursive neural network AdaRNN. The model analyzes the sentiment structure and context information to learn the sentiment relationships between target and other words in the sentence. Nguyen et al. [9] combined recursive neural network with dependency tree and presented PhraseRNN. The model builds a binary phrase dependency tree for features in different aspects and judges sentiment polarity of different aspects with the dependency tree. Tang et al. [10] proposed Target-Dependent Long Short-Term Memory (TD-LSTM). The model uses two LSTM network, which, respectively, propagate the pre-context and back propagate the post-context, to evaluate the target in the context.

In addition to target information analysis, attention mechanism is put forward to promote the accuracy of the aspect-level sentiment analysis in recent years [11]. The attention mechanism is an imitation of human's behavior that, when a man is reading some text, he tends to focus on some specific words or phrases that are important for understanding the text. Attention is a vector whose length is equal to the sentence, and the value of each dimension represents the importance rank of the corresponding word. The attention vector usually element-wise multiplies the sentence representation, which allows the model to focus on the important parts of the sentence. Wang et al. [11] proposed an attention-based LSTM network, who applies attention on the LSTM hidden layer through multiplication with aspect embedding vector. The model focuses on the aspect information and has made great progress in the ABSA tasks. Tang et al. [12] raised an attention-based memory network which has two attentions, content-level attention and aspect position attention. The two attentions enhance the model's ability to extract the features of targets from sentence with multi-aspect, which makes up the shortage of single attention models. Du et al. [13] introduced a CNN-based attention model. The model uses CNN's invariance feature to extract the attention vector, and has received better experiment results than LSTM-based attention implementations.

The above studies show that conventional NLP methods that focus on the sentence level features cannot meet the demand of aspect level sentiment classification. More specialized features, such as target of the sentence and people's attention on the sentence, should be analyzed. The approaches mentioned above tend to use one of these features to solve the problems and have made some great progress in the aspect level sentiment analysis domain. In our opinion, the combination of target information analysis and attention mechanism can further simulate human's activity and get better achievements in the domain. People tend to use idioms in the comments and attention mechanism can prevent the model from recognizing the target in the idioms by mistake. For instance, in the sentence "it is the service that leaves a bad taste in my mouth", the highlighted target by the attention mechanism should be "service". However, a non-attention model may also mark "taste" as a second target, when the idiom "leaves a bad taste in my mouth" actually means "leaves a bad impression" and does not mean the taste of the food is bad.

In this paper, we propose an attention-based aspect level model, AARCNN, to analyze the sentiment polarity at the aspect level. The model integrates Bi-LSTM network and convolutional neural network (CNN), which are, respectively, for target analysis and attention extraction.

The main contributions of our work are as follows:

1. We propose AARCNN, an integrated model for aspect-based sentiment analysis tasks. The model incorporates attention mechanism and aspect information, and uses the important part of the comment sentence to analyze the sentiment polarity in the aspects.
2. Both attention and aspect are essential in the ABSA tasks, so we introduce both modules in our model: CNN-based attention mechanism for attention extraction and weighted aspect embedding for aspect detection and analysis.

- Experimental results indicate that the approach achieves state-of-the-art performance compared to several baseline methods, and further examples prove that the substitution of Bi-LSTM promotes that classification accuracy.

The rest of this paper is structured as follow: Section 2 is the detailed description of AARCNN model. Section 3 shows the experimental results and corresponding case study of AARCNN model. Section 4 concludes our work.

2. Model

The architecture of AARCNN model is shown in Figure 1, which consists of five parts: input module, memory module, attention discover module, representation module and output module. Suppose the input comment sentence is $s = \{s_1, s_2, \dots, s_n\}$ where s_i is the i^{th} word of the sentence; the goal is to classify the sentiment polarity of each aspect of the comment. For example, given the comment “The food is not cheap but very delicious”, the output should indicate that it is negative in the “FOOD#PRICE” aspect and is positive in the “FOOD#TASTE” aspect.

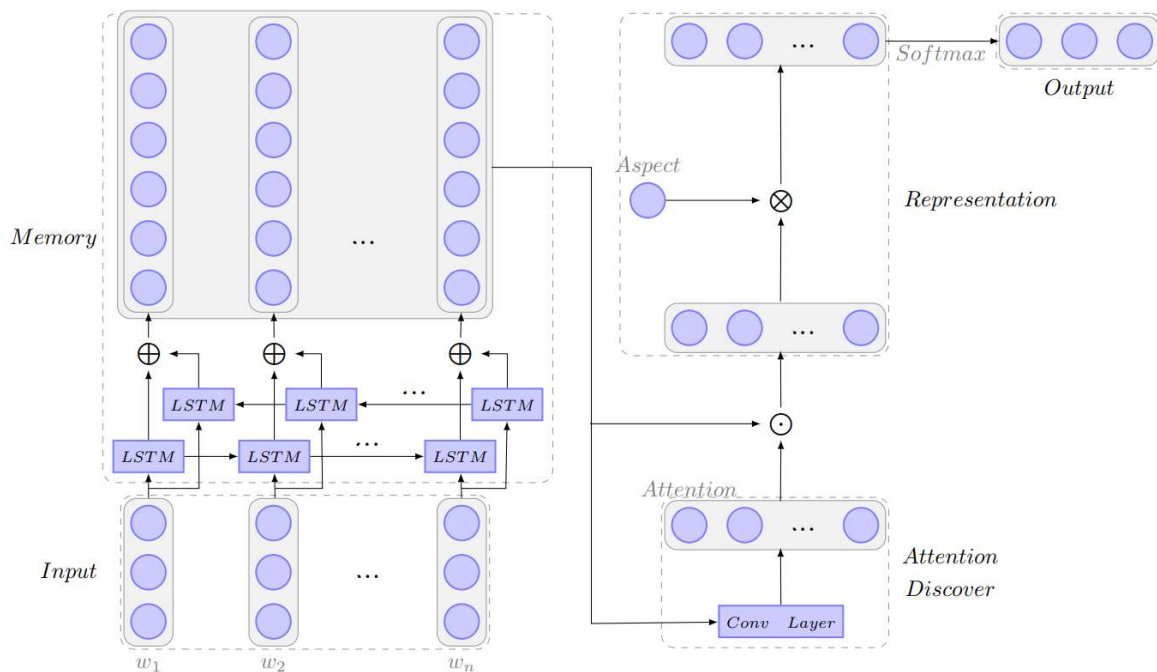


Figure 1. The architecture of AARCNN.

2.1. Input Preprocess

The first step is formalizing the input comment sentence. Suppose the sentence is composed of a n -word sequence. Each word is transferred to a real-value vector with word embedding methods such as GloVe [14] or CBOW [15]. Word embedding is an unsupervised method that provides a dictionary mapping words to their unique corresponding vector. Then, we can get a sequence of word embedding $S = [w_1, w_2, \dots, w_n]$ where $w_t \in R^d$ is the word vector of the t^{th} word and d is the dimension of vectors in the embedding dictionary. A pre-trained word embedding model is applied in our model.

2.2. Memory Module

The sequence is input into a Bi-LSTM network to learn the hidden semantics of the words. Since the words may have strong connection with its context even if the distance between two words is long, we regard that LSTM [16] is good at learning the long-term dependencies in this scenario. Especially, we mentioned that the comment sentences tend to have two kinds of expression order, e.g., “the spot is perfect” and “the perfect spot”. Considering the order affects the dependencies between

words, we replace LSTM with Bi-LSTM, which calculates the hidden semantic in both normal order and opposite order.

As shown in Figure 2, at each time step t , given the word vector w_t , the previous hidden state h_{t-1} and the previous cell state c_{t-1} , the current state can be calculated as follows:

$$i_t = \sigma(W_i w_t + U_i h_{t-1} + b_i) \quad (1)$$

$$f_t = \sigma(W_f w_t + U_f h_{t-1} + b_f) \quad (2)$$

$$o_t = \sigma(W_o w_t + U_o h_{t-1} + b_o) \quad (3)$$

$$\tilde{c}_t = \tanh(W_c w_t + U_c h_{t-1} + b_c) \quad (4)$$

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

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

where σ is the sigmoid function and \tanh is the hyperbolic tangent function. The symbol \cdot stands for matrix multiplication and \odot means element-wise multiplication. W s are the weight matrixes for the input vector w_t and U s are the weight matrixes for the previous hidden state vector h_{t-1} . i, f, o denote the input gate, forget gate and the output gate, which decide whether to update the cell state with the input, whether to forget the memory from the last time step and whether to output the memory, respectively.

The output of the forward LSTM is marked as \vec{h}_t . The backward LSTM uses the reversed sequence $S_r = [w_n, w_{n-1}, \dots, w_1]$ as the input and its output is remarked as \overleftarrow{h}_t , where t is the index of the word in the sequence S . The final representation of the sentence is $H = [h_1, h_2, \dots, h_n]$, where $h_t = [\vec{h}_t, \overleftarrow{h}_t]$.

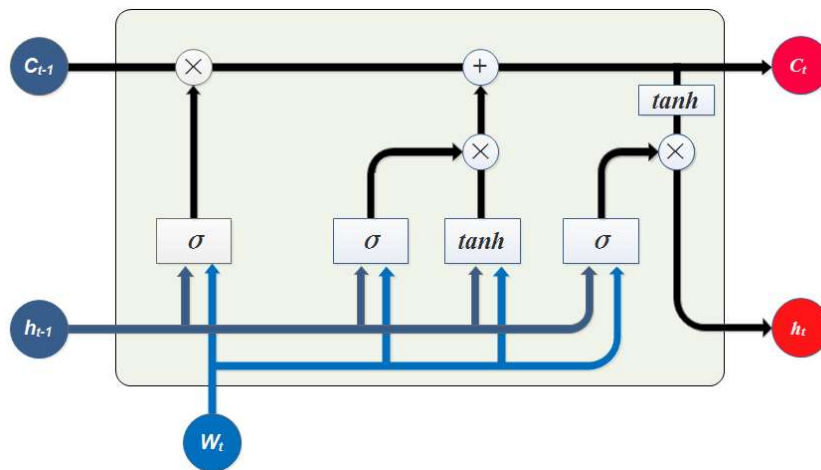


Figure 2. The architecture of LSTM cell used in the AARCNN model.

2.3. Attention Discover Module

In a comment sentence, only a few words do good to aspect detection and sentiment polarity judgement. A word or a phrase, which may show the aspect or the sentiment, is focused on when people try to understand the opinion, wherever it appears in the sentence. This feature is similar to translation invariance, a key feature of CNN, which leads to the application of CNN to find out the attention signal of the sentence.

Attention signal is a real-value vector that expresses the importance of a word in the sentence. Attention is marked as $AT = [at_1, at_2, \dots, at_n]$, where n is the length of the sentence and at_i is the score of the i^{th} word in the sentence; the higher score a word gets, the more important the word is in the sentence. The attention signal is calculated as the following steps:

Sentence representation padding. The first word of the sentence has no pre-context and the last word has no post-context, which may weaken their importance score when calculated in the same way as words in the middle. Therefore, the representation is filled with start vector and end vector as $H_s = [st^T, H, e^T]$, where $st^T \in R^{2d}$ and $e^T \in R^{2d}$ are pre-trained vector.

Scoring. We used the output of a CNN as the attention signal. To get a stable attention signal, the scoring module sets m filters in the convolution layer and uses their average as the final score. Let c_1, c_2, \dots, c_m be the output vector of filters f_1, f_2, \dots, f_m ; the attention signal is shown in Figure 3.

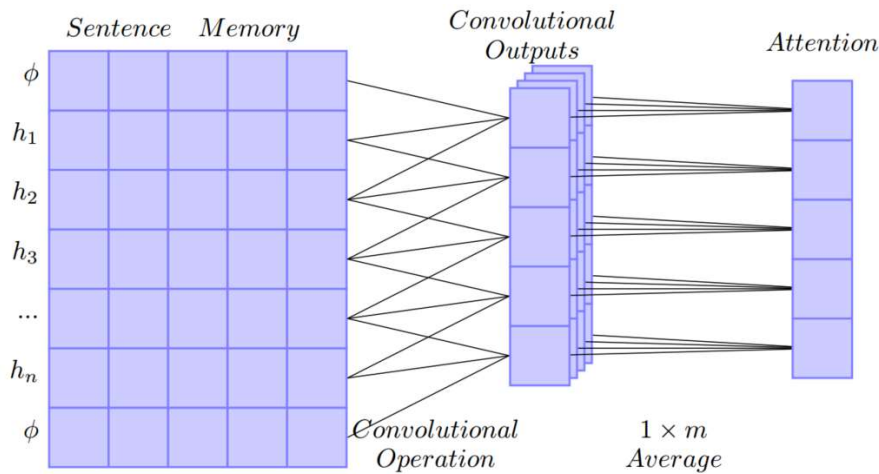


Figure 3. CNN-based model of attention signal extraction.

The attention signal highlights the important words in the sentence, which is applied through element-wise multiplication. Denote the sentence representation with attention by $r_\alpha = [r_{\alpha 1}, r_{\alpha 2}, \dots, r_{\alpha n}] \in R^{d \times n}$, where $r_{\alpha i} \in R^d$ which is computed as follow:

$$r_{\alpha i} = \alpha_i \times h_i \quad (7)$$

2.4. Aspect Embedding

Target information and aspects are important in the ABSA tasks. To use this information, we transfer the candidate targets and aspects to vectors through weighted aspect embedding. Weighted aspect embedding is finished as the following steps:

Given a target-aspect pair $T\#A$, both target and aspect word are put into the word embedding methods, and the output vectors are marked as v_t and v_a , where v_t and v_a are the embedding vectors for the target and aspect, respectively.

Each aspect embedding vector A_i is the weighted sum of v_t and v_a :

$$A_i = \alpha \cdot v_t + (1 - \alpha) \cdot v_a \quad (8)$$

where α is the adjustment factor for the formula, which controls the model's tendency to detect the target or the aspects.

All aspect embedding vectors are composed to a matrix $A = [A_1, A_2, \dots, A_s] \in R^{d \times s}$, where $a_i \in R^d$ is the word embedding vector for aspect word i , and s is the number of aspect embedding vectors.

2.5. Representation and Output Module

We multiply A and r_α , and get the aspect-level sentence representation R :

$$R = A^T \times r_\alpha \in R^{s \times n} \quad (9)$$

As the Bi-LSTM network has discovered implied aspect word, the row of R with aspect appearing will have a larger value. Each row of the final representation matrix represents the possibility of aspect that may be valid, which is vector of n dimension. If an aspect appears in some part of the sentence, the value in the corresponding dimension will be high. The model detects aspects through max-pooling, which calculates the highest value of each row and uses a threshold filter to get the filtered aspect-level sentence representation $R_f \in R^{k \times n}$ where k is the number of detected aspects.

The filtered aspect-level representation R_f is fed into a softmax layer to predict the sentiment polarity of each aspect:

$$y_k = \text{softmax}(W_s R_{fk} + b_s) \quad (10)$$

where R_{fk} is k^{th} row of R_f , and W_s and b_s are the parameters for softmax layer.

The cost function in our model is cross entropy, which is trained through back-propagation algorithm:

$$\text{cost} = - \sum_{i \in D} \sum_{j \in C} y_i^j \log \hat{y}_i^j + \lambda \|\theta\|^2 \quad (11)$$

where D is the training dataset, C is the sentiment category set, $y_i^j \in R^{|C|}$ is a one-hot vector which denotes the true sentiment polarity in the aspect, and $\hat{y}_i^j \in R^{|C|}$ is the prediction. L2 norm [17] is introduced in the cost function to avoid the weights in softmax being too large, where λ is the weight of L_2 regularization term, and $\|\theta\|^2$ is the sum of squares of the weight parameter in W_s . The model is trained by minimizing the cost and we adopt dropout [18,19] to avoid overfitting.

3. Experiment

3.1. Dataset

The dataset is from SemEval 2016 task5 [1], where we picked comments about *Restaurants* and *Cameras* to verify our model. There are two parts in each piece of datum in the training set: the comment text and its opinions. The opinions are composed of the sentiment polarity and the category in the form of "ENTITY#ASPECT". The restaurants dataset is an English dataset and camera is in Chinese. The detailed dataset statistics are shown in Table 1.

Table 1. The Summary statistics of the two datasets. Restaurants is an English dataset, whose sentiment polarity is positive, neutral and negative. Cameras is a Chinese dataset, whose polarity is positive and negative.

Dataset	Positive		Neutral		Negative		Total	
	Train	Test	Train	Test	Train	Test	Train	Test
Restaurants	1657	611	101	44	749	204	2507	859
Cameras	809	344	-	-	450	137	1259	481

3.2. Task Definition

Given a comment sentence and a set of aspects, the model first should recognize the target entities in the comment and their corresponding aspect. Then, the approach should predict the sentiment polarity of all "ENTITY#ASPECT" pair. Take a piece of sample datum in the restaurants dataset as an example; for the input sentence "For the price, you cannot eat this well in Manhattan", the model should output positive for "RESTAURANT#PRICES" and positive for "FOOD#QUALITY".

3.3. Model Training and Parameters

For the word embedding method in the input module and aspect embedding module, we pre-trained a GloVe [14] model. The training corpora was *Leipzig Corpora Collection* [20] and the dimension of embedding vector was set to 300. To avoid out-of-vocabulary (OOV) problems,

we sampled from a uniform distribution $U(-0.1, 0.1)$ for the words that were not recorded in the dictionary. The Chinese dataset was preprocessed by word segment with *Stanford Word Segmenter* [21].

For the CNN in the attention module, 100 filters were used to reduce the possibility of missing important signals. The CNN network contained one convolution layer, in which the filter was of $R^{3 \times 300}$ with no padding and the stride was 1.

We trained the model through mini-batch gradient descend, and the mini-batch was 32. L_2 regularization factor was added to the loss function and the weight for L_2 factor was 3. The dropout rate was set to 0.5 and the learning rate was set to 0.01.

3.4. Comparison with Baseline Methods

We set several baseline methods, including LSTM, TD-LSTM, AT-LSTM, AT-CNN and RCNN-LSTM.

LSTM: Standard LSTM [22] network is the improvement of conventional Recurrent Neural Network, which is also good at analyzing the dependencies between words in the sentence and has resolved the “long-dependency” problem.

TD-LSTM: TD-LSTM [10] adds target information analysis to the standard LSTM network. It regards aspect as a target, which enables it to classify the sentiment in each aspect.

AT-LSTM: AT-LSTM [11] applies attention mechanism to the LSTM network, so the model can focus on the important parts in the sentence. The model is more effective classifying the sentiment in each valid aspect.

AT-CNN: AT-CNN [23] is a CNN-based attention mechanism implementation model, which takes advantages of the invariance feature of CNN to rank the importance.

RCNN-LSTM: RCNN-LSTM [24] is a combined model with Regional Convolution Neural Network (RCNN) and LSTM network. The region works as the attention mechanism to capture the important parts and the RCNN outputs the attentive sentence representation. Through LSTM network, the representation is analyzed to get the final sentiment.

3.5. Result and Analysis

We performed experiment for AARCNN model and the five models in Section 4 with same training dataset and test dataset. Each model was trained and tested five times with the same training data in different input order. The results are shown in Table 2 as the average stats.

Table 2. Experimental data on Restaurants and Cameras dataset. Evaluation metrics are accuracy and F1 measure. Best scores are in bold.

Method	Restaurants		Cameras	
	Accuracy	F1	Accuracy	F1
LSTM	74.21	81.55	84.59	87.90
TD-LSTM	75.67	82.75	85.43	88.93
AT-LSTM	76.43	83.24	86.28	89.28
AT-CNN	76.08	83.11	85.97	89.23
RCNN-LSTM	78.59	83.58	86.32	89.41
AARCNN	79.04	84.00	86.75	89.80

In Table 2, we can see that LSTM gets the worst performance. Standard LSTM model only builds memory of the sentence and considers neither aspect information nor attention information. It fails to detect the sentiment for different aspects and only outputs an overall sentiment for the sentence. Therefore, standard LSTM network is not suitable for the ABSA task.

TD-LSTM beats LSTM by over 1% on both Restaurants and Cameras datasets, which shows that the aspect information does good to the ABSA task. However, TD-LSTM only introduces aspect information and does not contain attention mechanism, so it cannot recognize whether a word is

truly important to express the sentiment. Unimportant words may disturb the analysis, and may lead to mistakes.

The two attention-based approaches, AT-LSTM and AT-CNN, outperform LSTM by about 2%. The two models only introduce attention mechanism. Although the models cannot analyze the aspect information, they still show better performance than TD-LSTM, which means attention mechanism alone is better to promote the accuracy of ABSA task than aspect information analysis alone. Additionally, AT-CNN gets a higher score than AT-LSTM, which means that CNN-based attention model is better than LSTM-based attention model.

RCNN-LSTM gets the highest score in the baseline methods. Compared to AT-CNN, RCNN-LSTM adds LSTM network to model sentence representation, which assures the dependency information does not get lost. Compared to AT-LSTM, its independent attention module has better performance to extract the important parts in the sentence.

Our model, AARCNN, gets the best score. Our model takes advantages of both attention mechanism and aspect information to analyze the sentiment of the comment sentence. The model has promotion on the accuracy and F1 value, but it costs about twice the time compared to single LSTM network.

3.6. The Use of Bi-LSTM

Bi-LSTM network is used instead of standard LSTM network in our model to optimize the sentence with reversed expression. We do an experiment to find out the improvement Bi-LSTM has made to the model. Our model is marked as Bi-LSTM-* and the model with LSTM network as LSTM-*. The LSTM-* model was trained and tested with the same set with the one used in the baseline experiment. The result is shown in Table 3.

Table 3. Contrast experiment data of Bi-LSTM-* and LSTM-* on Restaurants and Cameras datasets. Evaluation metrics are accuracy and F1 measure. Best scores are in bold.

Models	Restaurants		Cameras	
	Accuracy	F1	Accuracy	F1
Bi-LSTM-*	79.04	84.00	86.75	89.80
LSTM-*	78.67	83.65	86.40	89.52

In Table 3, we can find that Bi-LSTM-* has an average 0.3% better performance than LSTM-* with both accuracy and F1 score on restaurants and camera datasets. We go deep into the data, and find that Bi-LSTM-* has almost the same accuracy as LSTM-* on most review sentences. However, the model with Bi-LSTM has about 3% higher performance than LSTM-* on long sentences where the target is in front of its description and the distance between them is more than two words. For example, in the review sentence in restaurants “We took advantage of the half price sushi deal on Saturday so it was well worth it”, the target “half price sushi deal” and its description “well worth” are four words away, which causes the LSTM-* to fail to recognize the aspect “QUALITY” and mark the sentence as “RESTAURANT#GENERAL”. As shown in Table 4, the sentences of this kind occupy about 10–15% in the test set, which leads to only 0.3% in the overall experiment result.

Table 4. The Summary statistics of inverted sentences on Restaurants and Cameras datasets. *Special* stands for the number of sentences where the target is in front of its description and the distance between them is more than two words.

Dataset	Special		Total	
	Train	Test	Train	Test
Restaurants	358	100	2507	859
Cameras	167	69	1259	481

The result proves the effect of Bi-LSTM cell in the model. However, Bi-LSTM-* model costs much more time than LSTM-* model, which is the main reason for the slow pace of model training.

Additionally, the test score of LSTM-* exceeds the best baseline method RCNN-LSTM, which demonstrates that the architecture of our model fully exploits each module in the ABSA tasks.

3.7. Case Study

We analyzed a comment sentence “The food was well prepared and the service impeccable” as a case study. The true annotation of the sentence is positive in FOOD#TASTE and positive in SERVICE#GENERAL. Besides the two true aspects, two wrong aspects are also chosen in the test, which are RESTAURANT#GENERAL and FOOD#PRICE. RESTAURANT#GENERAL is weakly relevant to the sentence and FOOD#PRICE is totally irrelevant to the sentence. Figure 4 shows the final representation of the sentence under the four circumstances. The red color in the figure is the signal of positive sentiment and the blue represents negative sentiment.

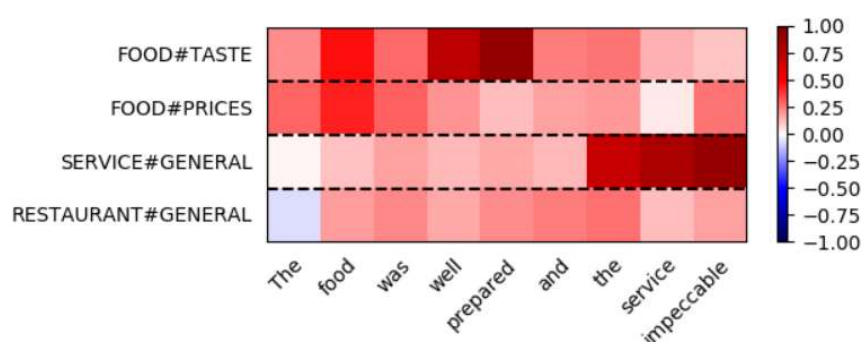


Figure 4. Final representation visualization. Each row represents the relationships between the corresponding aspect and the sentence. The darker color means the greater correlation of the sentence and the aspect.

In Figure 4, we can find that, to the first half of the sentence, the model gives high positive ranks to the FOOD#TASTE. Although the sentence is also relevant to the aspect FOOD#PRICES, its connection is not as strong. The second half is SERVICE#GENERAL, and the same reason as above has prevented the RESTAURANT#GENERAL from getting a high score. The irrelevant aspect FOOD#PRICES has the lowest score in each part of the sentence, which means the model does not choose it as a valid aspect. Because the model uses max-pooling as the filter to select aspects, both FOOD#TASTE and SERVICE#GENERAL that have top scores can pass through the filter, but the RESTAURANT#GENERAL with low scores in each half fails to pass it. The final representations of 1 and 3 is sent to the softmax classifier and the output of our model is positive in FOOD#TASTE and positive in SERVICE#GENERAL.

4. Conclusions

In this paper, a new model is proposed, AARCNN, to finish the aspect-based sentiment analysis tasks. The approach has integrated CNN-based attention extraction and aspect embedding to raise the accuracy of sentiment classification. The results of the experiments demonstrate that our model exceeds the conventional single neural network methods such as TD-LSTM and AT-LSTM and has made better use of different neural networks in the model than multi-neural network methods such as RCNN-LSTM. Moreover, the experimental results also prove that attention mechanism and aspect information analysis can work together to achieve greater promotion to the accuracy of the ABSA tasks. Additionally, we tested Bi-LSTM network in our model and found that Bi-LSTM network tends to have better performance than standard LSTM network when modeling natural language comment sentences, which may have reversed expressions.

Author Contributions: Conceptualization, S.L.; investigation, Y.Z.; writing—original draft preparation, X.G.; writing—review and editing, X.G. and W.Z.; project administration, Y.Z.; and funding acquisition, Y.Z. and Y.Z.

Funding: This research was funded by The National Key Research and Development Plan of China (Grant No. 2017YFD0400101) and Zhejiang Province medical and health science and technology platform project (Grant No. 2017KY497).

Acknowledgments: The authors thank the anonymous reviewers for their valuable and constructive comments.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Pontiki, M.; Galanis, D.; Papageorgiou, H.; Androutsopoulos, I.; Manandhar, S.; Al-Smadi, M.; Al-Ayyoub, M.; Zhao, Y.; Qin, B.; Clercq, O.D. Semeval-2016 task 5: Aspect based sentiment analysis. In Proceedings of the International Workshop on Semantic Evaluation, San Diego, CA, USA, 16–17 June 2016; pp. 19–30.
2. Ma, D.; Li, S.; Zhang, X.; Wang, H. Interactive attention networks for aspect-level sentiment classification. In Proceedings of the International Joint Conference on Artificial Intelligence, Melbourne, Australia, 19–25 August 2017; pp. 4068–4074.
3. Singh, V.K.; Piryani, R.; Uddin, A.; Waila, P. Sentiment analysis of movie reviews: A new feature-based heuristic for aspect-level sentiment classification. In Proceedings of the International Multi-Conference on Automation, Computing, Communication, Control and Compressed Sensing, Kottayam, India, 22–23 March 2013; pp. 712–717.
4. Joachims, T. Transductive inference for text classification using support vector machines. In Proceedings of the Sixteenth International Conference on Machine Learning, Bled, Slovenia, 27–30 June 1999; pp. 200–209.
5. McCallum, A.K.; McCallum, A.K.; Thrun, S.; Mitchell, T. Text classification from labeled and unlabeled documents using EM. *Mach. Learn.* **2000**, *39*, 103–134.
6. Jiang, L.; Yu, M.; Zhou, M.; Liu, X.; Zhao, T. Target-dependent twitter sentiment classification. In Proceedings of the Annual Meeting of the Association for Computational Linguistics Human Language Technologies, Portland, OR, USA, 19–24 June 2011; Volume 1, pp. 151–160.
7. Mikolov, T.; Karafiát, M.; Burget, L.; Cernocký, J.; Khudanpur, S. Recurrent neural network based language model. In Proceedings of the INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Chiba, Japan, 26–30 September 2010; pp. 1045–1048.
8. 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 Meeting of the Association for Computational Linguistics, Baltimore, MD, USA, 22–27 June 2014; pp. 49–54.
9. Nguyen, T.H.; Shirai, K. Phrasernn: Phrase recursive neural network for aspect-based sentiment analysis. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 2509–2514.
10. Tang, D.; Qin, B.; Feng, X.; Liu, T. Target-dependent sentiment classification with long short term memory. *arXiv* **2015**, arXiv:1512.01100.
11. Wang, Y.; Huang, M.; Zhu, X.; Zhao, L. Attention-based LSTM for aspect-level sentiment classification. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–4 November 2016; pp. 606–615.
12. Tang, D.; Qin, B.; Liu, T. Aspect level sentiment classification with deep memory network. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–4 November 2016; pp. 214–224.
13. Du, J.; Gui, L.; Xu, R.; He, Y. A convolutional attention model for text classification. In Proceedings of the National CCF Conference on Natural Language Processing and Chinese Computing, Dalian, China, 8–12 November 2017; pp. 183–195.
14. Pennington, J.; Socher, R.; Manning, C. Glove: Global vectors for word representation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
15. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.

16. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)] [[PubMed](#)]
17. Bilgic, B.; Chatnuntawech, I.; Fan, A.P.; Setsompop, K.; Cauley, S.F.; Wald, L.L.; Adalsteinsson, E. Fast image reconstruction with l2-regularization. *J. Magn. Reson. Imaging* **2014**, *40*, 181–191. [[CrossRef](#)] [[PubMed](#)]
18. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.
19. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
20. Goldhahn, D.; Eckart, T.; Quasthoff, U. Building large monolingual dictionaries at the Leipzig corpora collection: From 100 to 200 languages. In Proceedings of the 8th International Language Ressources and Evaluation (LREC'12), Istanbul, Turkey, 23–25 May 2012.
21. Tseng, H.; Chang, P.; Andrew, G.; Jurafsky, D.; Manning, C. A conditional random field word segmenter. *Found. Sci.* **2005**, 168–171.
22. Wang, X.; Liu, Y.; Sun, C.; Wang, B.; Wang, X. Predicting polarities of tweets by composing word embeddings with long short-term memory. In Proceedings of the Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing, Beijing, China, 26–31 July 2015; pp. 1343–1353.
23. Yin, W.; Schutze, H.; Xiang, B.; Zhou, B. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. *Trans. Assoc. Comput. Linguist.* **2016**, *4*, 259–272.
24. Wang, J.; Yu, L.C.; Lai, K.R.; Zhang, X. Dimensional sentiment analysis using a regional CNN-LSTM model. In Proceedings of the Meeting of the Association for Computational Linguistics, Berlin, Germany, 7–12 August 2016; pp. 225–230.



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).