Aspect Level Sentiment Classification with Attention-over-Attention Neural Networks

Binxuan Huang, Yanglan Ou and Kathleen M. Carley

Carnegie Mellon University,
5000 Forbe Ave., Pittsburgh, United States
{binxuanh,kathleen.carley}@cs.cmu.edu,yanglano@andrew.cmu.edu

Abstract. Aspect-level sentiment classification aims to identify the sentiment expressed towards some aspects given context sentences. In this paper, we introduce an attention-over-attention (AOA) neural network for aspect level sentiment classification. Our approach models aspects and sentences in a joint way and explicitly captures the interaction between aspects and context sentences. With the AOA module, our model jointly learns the representations for aspects and sentences, and automatically focuses on the important parts in sentences. Our experiments on laptop and restaurant datasets demonstrate our approach outperforms previous LSTM-based architectures.

1 Introduction

Unlike document level sentiment classification task [4,15], aspect level sentiment classification is a more fine-grained classification task. It aims at identifying the sentiment polarity (e.g. positive, negative, neutral) of one specific aspect in its context sentence. For example, given a sentence "great food but the service was dreadful" the sentiment polarity for aspects "food" and "service" are positive and negative respectively.

Aspect sentiment classification overcomes one limitation of document level sentiment classification when multiple aspects appear in one sentence. In our previous example, there are two aspects and the general sentiment of the whole sentence is mixed with positive and negative polarity. If we ignore the aspect information, it is hard to determine the polarity for a specified target. Such error commonly exists in the general sentiment classification tasks. In one recent work, Jiang et al. manually evaluated a Twitter sentiment classifier and showed that 40% of sentiment classification errors are because of not considering targets [6].

Many methods have been proposed to deal with aspect level sentiment classification. The typical way is to build a machine learning classifier by supervised training. Among these machine learning-based approaches, there are mainly two different types. One is to build a classifier based on manually created features [6,26]. The other type is based on neural networks using end-to-end training without any prior knowledge [11,25,28]. Because of its capacity of learning representations from data without feature engineering, neural networks are becoming popular in this task.

Because of advantages of neural networks, we approach this aspect level sentiment classification problem based on long short-term memory (LSTM) neural networks. Previous LSTM-based methods mainly focus on modeling texts separately [23,28], while

our approach models aspects and texts simultaneously using LSTMs. Furthermore, the target representation and text representation generated from LSTMs interact with each other by an attention-over-attention (AOA) module [2]. AOA automatically generates mutual attentions not only from aspect-to-text but also text-to-aspect. This is inspired by the observation that only few words in a sentence contribute to the sentiment towards an aspect. Many times, those sentiment bearing words are highly correlated with the aspects. In our previous example, there are two aspects "appetizers" and "service" in the sentence "the appetizers are ok, but the service is slow." Based on our language experience, we know that the negative word "slow" is more likely to describe "service" but not the "appetizers". Similarly, for an aspect phrase, we also need to focus on the most important part. That is why we choose AOA to attend to the most important parts in both aspect and sentence. Compared to previous methods, our model performs better on the laptop and restaurant datasets from SemEval 2014 [17]

2 Related work

Sentiment Classification
Sentiment classification aims at detecting the sentiment polarity for text. There are various approaches proposed for this research question [12]. Most existing works use machine learning algorithms to classify texts in a supervision fashion. Algorithms like Naive Bayes and Support Vector Machine(SVM) are widely used in this problem [10, 15, 27]. The majority of these approaches either rely on n-gram features or manually designed features. Multiple sentiment lexicons are built for this purpose [14, 18, 22].

In the recent years, sentiment classification has been advanced by neural networks significantly. Neural network based approaches automatically learn feature representations and do not require intensive feature engineering. Researchers proposed a variety of neural network architectures. Classical methods include Convolutional Neural Networks [7], Recurrent Neural Networks [9, 24], Recursive Neural Networks [19, 29]. These approaches have achieved promising results on sentiment analysis.

Aspect Level Sentiment Classification
Aspect level sentiment classification is a branch of sentiment classification, the goal of which is to identify the sentiment polarity of one specific aspect in a sentence. Some early works designed several rule based models for aspect level sentiment classification, such as [3, 13]. Nasukawa et al. first perform dependency parsing on sentences, then they use predefined rules to determine the sentiment about aspects [13]. Jiang et al. improve the target-dependent sentiment classification by creating several target-dependent features based on the sentences' grammar structures [6]. These target-dependent features are further fed into an SVM classifier along with other content features.

Later, kinds of neural network based methods were introduced to solve this aspect level sentiment classification problem. Typical methods are based on LSTM neural networks. TD-LSTM approaches this problem by developing two LSTM networks to model the left and right contexts for an aspect target [23]. This method uses the last hidden states of these two LSTMs for predicting the sentiment. In order to better capture the important part in a sentence, Wang et al. use an aspect term embedding to generate an attention vector to concentrate on different parts of a sentence [28]. Along these

lines, Ma et al. use two LSTM networks to model sentences and aspects separately [11]. They further use the hidden states generated from sentences to calculate attentions to aspect targets by a pooling operation, and vice versa. Hence their IAN model can attend to both the important parts in sentences and targets. Their method is similar to ours. However, the pooling operation will ignore the interaction among word-pairs between sentences and targets, and experiments show our method is superior to their model.

3 Method

Problem Definition

In this aspect level sentiment classification problem, we are given a sentence $s = [w_1, w_2, ..., w_i, .., w_j, ..., w_n]$ and an aspect target $t = [w_i, w_{i+1}, ..., w_{i+m-1}]$. The aspect target could be a single word or a long phrase. The goal is to classify the sentiment polarity of the aspect target in the sentence.

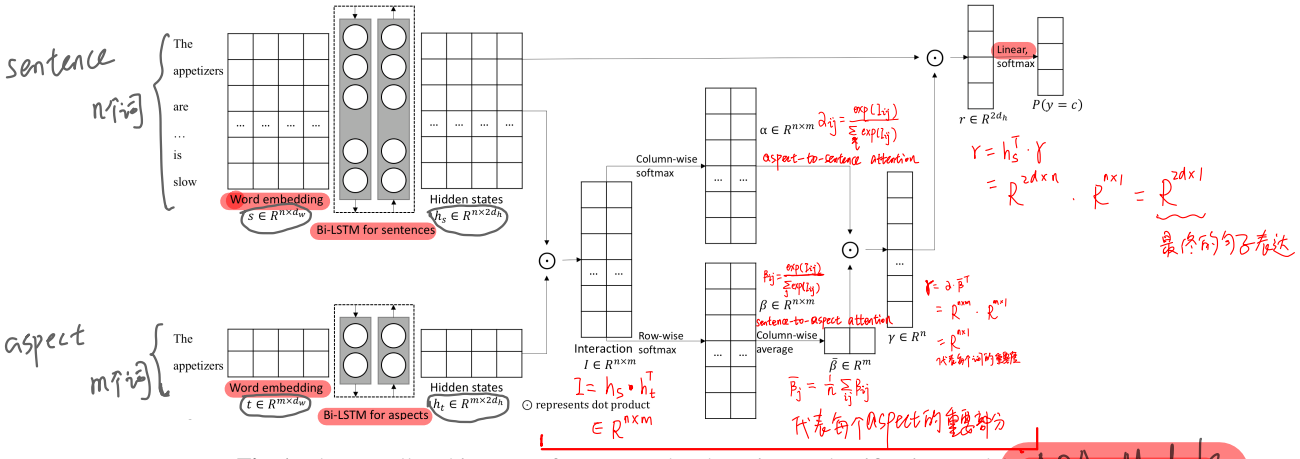

Fig. 1. The overall architecture of our aspect level sentiment classification model.

The overall architecture of our neural model is shown in Figure 1. It is mainly composed of four components: word embedding, Bidirectional-Long short-term memory (Bi-LSTM), Attention-over-Attention module and the final prediction.

Word Embedding

Given a sentence $s = [w_1, w_2, ..., w_i, .., w_j, ..., w_n]$ with length n and a target $t = [w_i, w_{i+1}, ..., w_{i+m-1}]$ with length m, we first map each word into a low-dimensional real-value vector, called word embedding [1]. For each word w_i, we can get a vector $v_i \in R^{d_w}$ from $M^{V \times d_w}$, where V is the vocabulary size and d_w is the embedding dimension. After an embedding look up operation, we get two sets of word vectors $[v_1; v_2; ...; v_n] \in R^{n \times d_w}$ and $[v_i; v_{i+1}; ...; v_{i+m-1}] \in R^{m \times d_w}$ for the sentence and aspect phrase respectively.

Bi-LSTM

After getting the word vectors, we feed these two sets of word vectors into two Bidirectional-

LSTM networks respectively. We use these two Bi-LSTM networks to learn the hidden semantics of words in the sentence and the target. Each Bi-LSTM is obtained by stacking two LSTM networks. The advantage of using LSTM is that it can avoid the gradient vanishing or exploding problem and is good at learning long-term dependency [5].

With an input $s = [v_1; v_2; ...; v_n]$ and a forward LSTM network , we generate a sequence of hidden states $\overrightarrow{h_s} \in R^{n \times d_h}$, where d_h is the dimension of hidden states. We generate another state sequence $\overleftarrow{h_s}$ by feeding s into another backward LSTM. In the Bi-LSTM network, the final output hidden states $h_s \in R^{n \times 2d_h}$ are generated by concatenating $\overrightarrow{h_s}$ and $\overleftarrow{h}_s$. We compute the hidden semantic states h_t for the aspect target t in the same way.

$$\overrightarrow{h_s} = \overrightarrow{LSTM}([v_1; v_2; ...; v_n]) \tag{1}$$

$$\overleftarrow{h_s} = \overleftarrow{LSTM}([v_1; v_2; ...; v_n]) \tag{2}$$

$$h_s = [\overrightarrow{h_s}, \overleftarrow{h_s}] \tag{3}$$

Attention-over-Attention

Given the hidden semantic representations of the text and the aspect target generated by Bi-LSTMs, we calculate the attention weights for the text by an AOA module. This is inspired by the use of AOA in question answering [2]. Given the target representation $h_t \in R^{m \times 2d_h}$ and sentence representation $h_s \in R^{n \times 2d_h}$, we first calculate a pair-wise interaction matrix $I = h_s \cdot h_t^T$, where the value of each entry represents the correlation of a word pair among sentence and target. With a column-wise softmax and row-wise softmax, we get target-to-sentence attention α and sentence-to-target attention β. After column-wise averaging β, we get a target-level attention $\bar{\beta} \in R^m$, which indicating the important parts in an aspect target. The final sentence-level attention $\gamma \in R^n$ is calculated by a weighted sum of each individual target-to-sentence attention α, given by equation (7). By considering the contribution of each aspect word explicitly, we learn the important weights for each word in the sentence.

$$\alpha_{ij} = \frac{exp(I_{ij})}{\sum_i exp(I_{ij})} \tag{4}$$

$$\beta_{ij} = \frac{exp(I_{ij})}{\sum_j exp(I_{ij})} \tag{5}$$

$$\bar{\beta}_j = \frac{1}{n} \sum_i \beta_{ij} \tag{6}$$

$$\gamma = \alpha \cdot \bar{\beta}^T \tag{7}$$

Final Classification

The final sentence representation is a weighted sum of sentence hidden semantic states using the sentence attention from AOA module.

$$r = h_s^T \cdot \gamma \tag{8}$$

We regard this sentence representation as the final classification feature and feed it into a linear layer to project r into the space of targeted C classes.

$$x = W_l \cdot r + b_l \tag{9}$$

where W_l and b_l are the weight matrix and bias respectively. Following the linear layer, we use a softmax layer to compute the probability of the sentence s with sentiment polarity $c \in C$ towards an aspect a as:

$$P(y = c) = \frac{exp(x_c)}{\sum_{i \in C} exp(x_i)} \tag{10}$$

The final predicted sentiment polarity of an aspect target is just the label with the highest probability. We train our model to minimize the cross-entropy loss with L_2 regularization

$$loss = -\sum_i \sum_{c \in C} I(y_i = c) \cdot log(P(y_i = c)) + \lambda ||\theta||^2 \tag{11}$$

where $I(\cdot)$ is an indicator function. λ is the L_2 regularization parameter and θ is a set of weight matrices in LSTM networks and linear layer. We further apply dropout to avoid overfitting, where we randomly drop part of inputs of LSTM cells.

We use mini-batch stochastic gradient descent with Adam [8] update rule to minimize the loss function with respect to the weight matrices and bias terms in our model.

4 Experiments

Dataset

We experiment on two domain-specific datasets for laptop and restaurant from SemEval 2014 Task 4 [26]. Experienced annotators tagged the aspect terms of the sentences and their polarities. Distribution by sentiment polarity category are given in Table 1.

Dataset	Positive	Neutral	Negative
Laptop-Train	994	464	870
Laptop-Test	341	169	128
Restaurant-Train	2164	637	807
Restaurant-Test	728	196	196

Table 1. Distribution by sentiment polarity category of the datasets from SemEval 2014 Task 4. Numbers in table represent numbers of sentence-aspect pairs.

Hyperparameters Setting

In experiments, we first randomly select 20% of training data as validation set to tune the hyperparameters. All weight matrices are randomly initialized from uniform distribution $U(-10^{-4}, 10^{-4})$ and all bias terms are set to zero. The L_2 regularization coefficient is set to 10^{-4} and the dropout keep rate is set to 0.2[20]. The word embeddings are initialized with 300-dimensional Glove vectors [16] and are fixed during training.

For the out of vocabulary words we initialize them randomly from uniform distribution $U(-0.01, 0.01)$. The dimension of LSTM hidden states is set to 150. The initial learning rate is 0.01 for the Adam optimizer. If the training loss does not drop after every three epochs, we decrease the learning rate by half. The batch size is set as 25.

Model Comparisons

We train and evaluate our model on these two SemEval datasets separately. We use accuracy metric to measure the performance. In order to further validate the performance of our model, we compare it with several baseline methods. We list them as follows:

Majority is a basic baseline method, which assigns the largest sentiment polarity in the training set to each sample in the test set.

LSTM uses one LSTM network to model the sentence, and the last hidden state is used as the sentence representation for the final classification.

TD-LSTM uses two LSTM networks to model the preceding and following contexts surrounding the aspect term. The last hidden states of these two LSTM network are concatenated for predicting the sentiment polarity [23].

AT-LSTM first models the sentence via a LSTM model. Then it combines the hidden states from the LSTM with the aspect term embedding to generate the attention vector. The final sentence representation is the weighted sum of the hidden states [28].

ATAE-LSTM further extends AT-LSTM by appending the aspect embedding into each word vector [28].

IAN uses two LSTM networks to model the sentence and aspect term respectively. It uses the hidden states from the sentence to generate an attention vector for the target, and vice versa. Based on these two attention vectors, it outputs a sentence representation and a target representation for classification [11].

Methods	Restaurant	Laptop
Majority	0.535	0.650
LSTM	0.743	0.665
TD-LSTM [23]	0.756	0.681
AT-LSTM [28]	0.762	0.689
ATAE-LSTM [28]	0.772	0.687
IAN [11]	0.786	0.721
AOA-LSTM	**0.812** (0.797$\pm$0.008)	**0.745** (0.726$\pm$0.008)

Table 2. Comparison results. For our method, we run it 10 times and show "best (mean$\pm$std)". Performance of baselines are cited from their original papers.

In our implementation, we found that the performance fluctuates with different random initialization, which is a well-known issue in training neural networks [21]. Hence, we ran our training algorithms 10 times, and report the average accuracy as well as the best one we got in Table 2. All the baseline methods only reported a single best number in their papers. On average, our algorithm is better than these baseline methods and our best trained model outperforms them in a large margin.

Case Study

In Table 3, We list five examples from the test set. To analyze which word contributes the most to the aspect sentiment polarity, we visualize the final sentence attention vectors γ in Table 3. The color depth indicates the importance of a word in a sentence, the darker

the more important. In the first two examples, there are two aspects "appetizers" and "service" in the sentence "the appetizers are ok, but the service is slow." We can observe that when there are two aspects in the sentence, our model can automatically point to the right sentiment indicating words for each aspect. Same thing also happens in the third and fourth examples. In the last example, the aspect is a phrase "boot time." From the sentence content "boot time is super fast, around any where from 35 seconds to 1 minute," this model can learn "time" is the most important word in the aspect, which further helps it find out the sentiment indicating part "super fast."

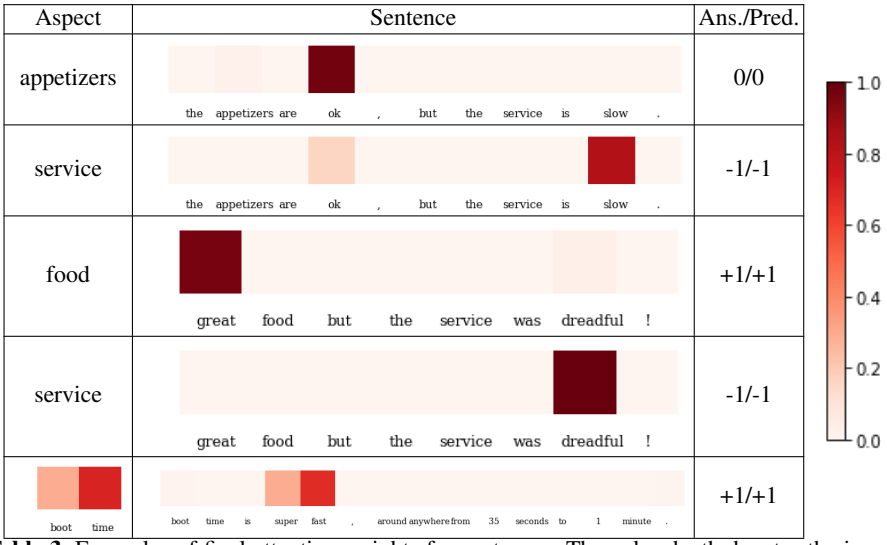

Table 3. Examples of final attention weights for sentences. The color depth denotes the importance degree of the weight in attention vector γ.

Error Analysis

The first type of major errors comes from non-compositional sentiment expression which also appears in previous works [25]. For example, in the sentence "it took about 2 1/2 hours to be served our 2 courses," there is no direct sentiment expressed towards the aspect "served." Second type of errors is caused by idioms used in the sentences. Examples include "the service was on point - what else you would expect from a ritz?" where "service" is the aspect word. In this case, our model cannot understand the sentiment expressed by idiom "on point." The third factor is complex sentiment expression like "i have never had a bad meal (or bad service) @ pigalle." Our model still misunderstands the meaning this complex expressions, even though it can handle simple negation like "definitely not edible" in sentence "when the dish arrived it was blazing with green chillis, definitely not edible by a human".

5 Conclusion

In this paper, we propose a neural network model for aspect level sentiment classification. Our model utilizes an Attention-over-Attention module to learn the important

parts in the aspect and sentence, which generates the final representation of the sentence. Experiments on SemEval 2014 datasets show superior performance of our model when compared to those baseline methods. Our case study also shows that our model learns the important parts in the sentence as well as in the target effectively.

In our error analysis, there are cases that our model cannot handle efficiently. One is the complex sentiment expression. One possible solution is to incorporate sentences' grammar structures into the classification model. Another type of error comes from uncommon idioms. In future work, we would like to explore how to combine prior language knowledge into such neural network models.

References

1. Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C.: A neural probabilistic language model. Journal of machine learning research 3(Feb), 1137–1155 (2003)
2. Cui, Y., Chen, Z., Wei, S., Wang, S., Liu, T., Hu, G.: Attention-over-attention neural networks for reading comprehension. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. pp. 593–602 (2017)
3. Ding, X., Liu, B.: The utility of linguistic rules in opinion mining. In: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 811–812. ACM (2007)
4. Glorot, X., Bordes, A., Bengio, Y.: Domain adaptation for large-scale sentiment classification: A deep learning approach. In: Proceedings of the 28th international conference on machine learning (ICML-11). pp. 513–520 (2011)
5. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation 9(8), 1735–1780 (1997)
6. 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-Volume 1. pp. 151–160. Association for Computational Linguistics (2011)
7. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1746–1751. Association for Computational Linguistics (2014)
8. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. In: Proceedings of the 3rd International Conference on Learning Representations (ICLR) (2015)
9. Lai, S., Xu, L., Liu, K., Zhao, J.: Recurrent convolutional neural networks for text classification. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence. vol. 333, pp. 2267–2273 (2015)
10. Liu, B., Blasch, E., Chen, Y., Shen, D., Chen, G.: Scalable sentiment classification for big data analysis using naive bayes classifier. In: Big Data, 2013 IEEE International Conference on. pp. 99–104. IEEE (2013)
11. Ma, D., Li, S., Zhang, X., Wang, H.: Interactive attention networks for aspect-level sentiment classification. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17. pp. 4068–4074 (2017)
12. Medhat, W., Hassan, A., Korashy, H.: Sentiment analysis algorithms and applications: A survey. Ain Shams Engineering Journal 5(4), 1093–1113 (2014)
13. Nasukawa, T., Yi, J.: Sentiment analysis: Capturing favorability using natural language processing. In: Proceedings of the 2nd international conference on Knowledge capture. pp. 70–77. ACM (2003)

14. Neviarouskaya, A., Prendinger, H., Ishizuka, M.: Sentiful: Generating a reliable lexicon for sentiment analysis. In: Affective Computing and Intelligent Interaction and Workshops, 2009. ACII 2009. 3rd International Conference on. pp. 1–6. IEEE (2009)
15. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: sentiment classification using machine learning techniques. In: Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10. pp. 79–86. Association for Computational Linguistics (2002)
16. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
17. Pontiki, M., Galanis, D., Papageorgiou, H., Androutsopoulos, I., Manandhar, S., AL-Smadi, M., Al-Ayyoub, M., Zhao, Y., Qin, B., De Clercq, O., et al.: Semeval-2016 task 5: Aspect based sentiment analysis. In: ProWorkshop on Semantic Evaluation (SemEval-2016). pp. 19–30. Association for Computational Linguistics (2016)
18. Qiu, G., Liu, B., Bu, J., Chen, C.: Expanding domain sentiment lexicon through double propagation. In: Proceedings of the 21st International Jont Conference on Artifical Intelligence. vol. 9, pp. 1199–1204 (2009)
19. Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C.D., Ng, A., Potts, C.: Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of the 2013 conference on empirical methods in natural language processing. pp. 1631–1642 (2013)
20. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research 15(1), 1929–1958 (2014)
21. Sutskever, I., Martens, J., Dahl, G., Hinton, G.: On the importance of initialization and momentum in deep learning. In: International conference on machine learning. pp. 1139–1147 (2013)
22. Taboada, M., Brooke, J., Tofiloski, M., Voll, K., Stede, M.: Lexicon-based methods for sentiment analysis. Computational linguistics 37(2), 267–307 (2011)
23. Tang, D., Qin, B., Feng, X., Liu, T.: Effective lstms for target-dependent sentiment classification. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers. pp. 3298–3307 (2016)
24. Tang, D., Qin, B., Liu, T.: Document modeling with gated recurrent neural network for sentiment classification. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 1422–1432 (2015)
25. 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)
26. Wagner, J., Arora, P., Cortes, S., Barman, U., Bogdanova, D., Foster, J., Tounsi, L.: Dcu: Aspect-based polarity classification for semeval task 4 (2014)
27. Wang, S., Manning, C.D.: Baselines and bigrams: Simple, good sentiment and topic classification. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2. pp. 90–94. Association for Computational Linguistics (2012)
28. Wang, Y., Huang, M., Zhu, X., Zhao, L.: Attention-based lstm for aspect-level sentiment classification. In: EMNLP. pp. 606–615 (2016)
29. Zhu, X., Sobihani, P., Guo, H.: Long short-term memory over recursive structures. In: International Conference on Machine Learning. pp. 1604–1612 (2015)