High Interpretable Transfer Network for Aspect Level Sentiment Classification

Anonymous ACL submission

Abstract

Aspect-level affective classification (ASC) aims to detect the affective polarity of a given viewpoint target in a sentence. In the ASC method based on neural network, most of the work uses the attention mechanism to capture the sentiment words corresponding to the opinion target, and then gather them as evidence to infer the emotion of the target. However, due to the complexity of annotation, the scale of aspect level data sets is relatively small. Data scarcity leads to the attention mechanism sometimes unable to pay attention to the sentiment words corresponding to the target, which finally weakens the performance of the neural model. In order to solve this problem, this paper proposes a complete High Interpretable Transfer Network transfer learning framework (HITN), which adopts methods such as data enhancement, attention adjustment and transfer to effectively improve the performance of ASC model. A large number of experimental results show that our method has always been all the previous migration methods in this field, even compared with some complex models.

1 Introduction

Aspect-level sentiment classification (ASC) is a fundamental subtask in sentiment analysis (Pontiki et al., 2014). Given a sentence and a opinion target (also called an aspect term) occurring in the sentence, aspect-level sentiment classification aims to infer the sentiment polarity in the sentence towards the target aspect. An opinion target, also known as aspect term, refers to a word or a phrase in review describing an aspect of an entity. For example, the electronic product comment "*The **speed** is fast, but the **screen** is dark*" consists of two aspect terms, namely "*speed*" and "*screen*". and they are associated with positive and negative sentiment respectively.

Traditional methods usually first artificially define a set of features, such as word bags, and then use machine learning methods to train a classifier (such as SVM) (Jiang et al., 2011). These methods depend on artificially defined features, and also need rich priori knowledge, such as constructing an sentiment dictionary. In recent years, with the development of deep learning technology, a number of neural network models (Tang et al., 2016) have been proposed and used in ASC tasks. Usually, these models use supervised learning to train classifiers, so a large amount of labeled data is necessary to obtain promising results. The cost of a large number of annotations is unbearable.

The number of training samples of the existing ASC publicly available data sets is very limited, which limits the performance of the neural network model. As a contrast, online websites such as Yelp contain a huge number of comments. These comments are usually accompanied by a rating score, from one star to five stars, which can intuitively show the user's satisfaction with something. After some simple preprocessing, these comments can be used as training data for document sentiment classification.

Generally speaking, the model for ASC task inputs a sentence and its contained aspect term to output sentiment category, while the model for DSC task inputs a sentence and outputs sentiment category. It can be seen that ASC task and DSC task are actually highly similar. Considering that ASC task lacks data and DSC task has a large number of available training data, a very direct idea is to use DSC data to improve the training of ASC model.

In view of this, He et al., 2018 proposed the PRET+MULT method to improve the shared embedded layer and LSTM layer by combining ASC task training and DSC task training. Similarly,

Chen et al., (2019) also adopted multi task joint training to improve their shared capsule layer. Different from the previous two methods, Zhao et al.,(2020) proposed ATN tried to transfer the attention learned from DSC model to ASC task. The above three works have good performance, but they have some defects in data preprocessing and interpretability, which we will explain in detail in our model part. In this paper, we propose a novel knowledge transfer framework, High Interpretable Transfer Network (HITN), which is more perfect and interpretable. We have made detailed and reliable explanations from data preprocessing to attention transfer. Compared with these three models, we have achieved better results, which also shows the effectiveness of our method. At the same time, it is worth noting that in order to focus on knowledge transfer, our network model structure is not complex. Nevertheless, we still achieved very good results.

In our framework, according to the inherent characteristics of ASC data, we firstly propose a novel DSC data preprocess method, which improves the disadvantage of large amount but insufficient diversity of DSC data, and makes DSC data closer to ASC data in terms of data distribution. In this section, we will also explain the defects of the previous three models in DSC data preprocess. Secondly, inspired by the multi-instance learning method (Ma et al., 2018), we use a simple LSTM model to train the attention weight and sentiment polarity score of each word in the text on the DSC data after data preprocess. Thirdly, we process the text attention distribution obtained in the DSC model to obtain an attention distribution more suitable for the characteristics of ASC tasks. Finally, we take the processed attention distribution as a priori knowledge and inject it into the training of ASC model. We conducted experiments on two SemEval datasets. The final results show that our method can be significantly improved by combining the two attention transfer methods, and is superior to all comparison methods in ASC tasks.

2 Related Work

2.1 Aspect-level Sentiment Classification

Early sentiment classification methods focused on document level sentiment classification tasks (DSC). Later, more fine-grained aspect sentiment classification tasks were proposed separately. At first, most ASC models define features manually, and then use machine learning methods for classification. The definition of these features is very dependent on expert experience and usually takes a lot of time and energy to obtain (Wang et al.,2004).

In recent years, deep learning methods have been widely used in various studies, and ASC is certainly no exception (Tang et al., 2016; Chen et al., 2017). Among these methods, neural network models widely used mainly include: LSTM based , CNN based , GNN based (Tang et al., 2020) and transformer based. Especially in recent years, many models have achieved excellent results by combining dependency tree of the sentence with GNN model.

However, for ASC tasks, no matter how complex the model is, it is difficult to mine enough information to help the model classify correctly in front of limited data. On the other hand, even a simple model can achieve good results after using a large amount of data. Different from many previous models that constantly try to improve the network structure, our HITN uses rich DSC annotation data to help ASC model improve the effect with simple network models.

2.2 Transfer Learning

Transfer learning aims to extract knowledge from one or more source tasks and then apply it to the target task. Previously, in the field of image processing, transfer learning has been proved to be effective. He et al., 2018 was the first person to transfer knowledge from document level data to improve ASC tasks by sharing the embedded layer and LSTM layer. Similar to their approach, Chen et al., 2019 used a capsule network to share bottom features between ASC and DSC tasks. Zhao et al., 2020 guided the training of ASC model by transferring attention. In this paper, our goal is to mine useful information from DSC data as much as possible according to the characteristics of ASC task. Therefore, from data preprocessing to training model design, we have fully considered the respective characteristics of ASC task and DSC task. Finally, experiments show that our method has better performance and better interpretability than the previous three models.

3 HITN

Figure 1 shows the overall architecture of the High Interpretable Transfer Network (HITN). It mainly consists of four parts: the DSC data

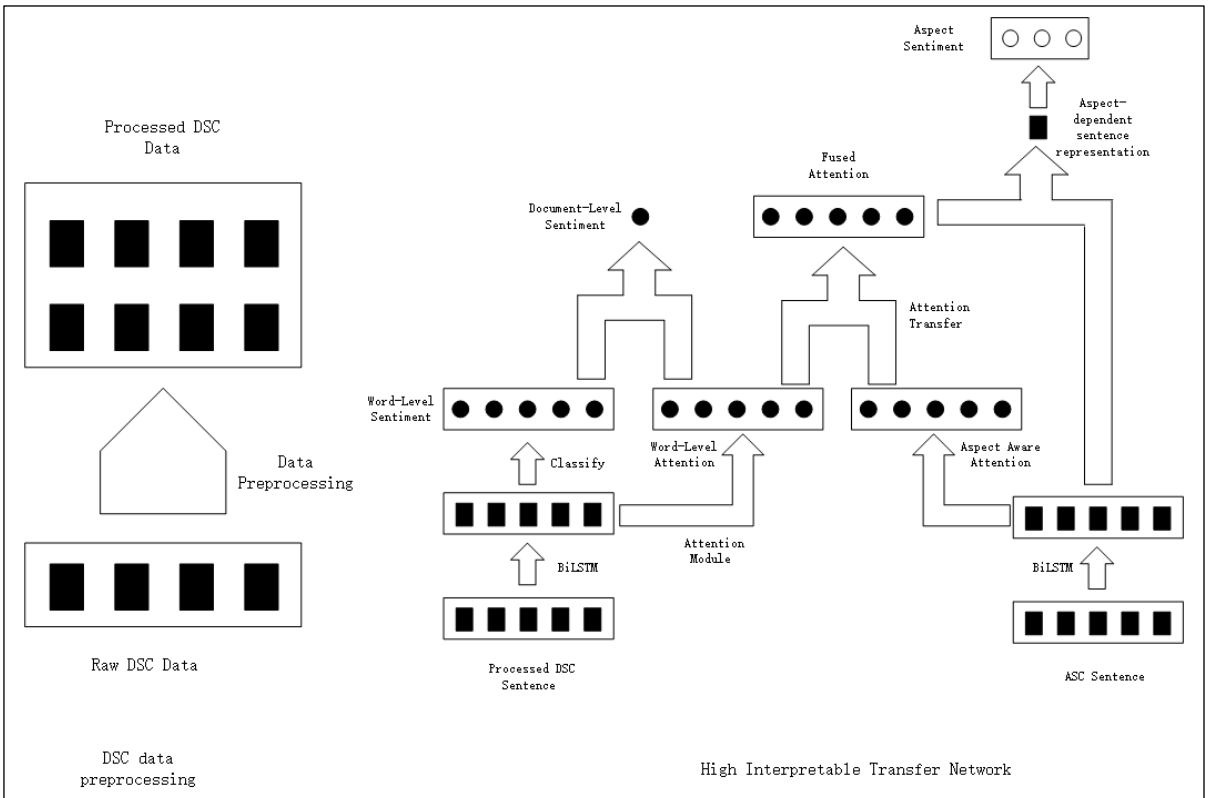

Figure 1: The overall architecture of HITN, which consists of four parts, DSC data preprocessing module, pre-trained DSC module, ASC module, and attention transfer module.

preprocessing module, the pre-trained DSC module, the ASC module, and the attention transfer module. In this section, we will first give the task formalization of ASC and DSC, then introduce the DSC data preprocessing module, the pre-trained DSC module, the ASC module and the attention transfer module.

3.1 Task Formalization

DSC Formalization For a review document d from the DSC dataset D, we regard it as a long sentence $\{w_1^d, w_2^d, \dots, w_n^d\}$ consisting of n words. DSC aims to determine the overall sentiment polarity of the review document d.

ASC Formalization Formally, given a sample < s, t > from the ASC dataset A, $s = \{w_1, w_2, \dots, w_n\}$ is a review sentence consisting of n words and $t = \{w_l, w_{l+1}, \dots, w_r\}$ is a given aspect term containing $|r - l|$ words. The aspect term t is a continuous subsequence of s. The goal of ASC is to predict the sentiment polarity (i.e., positive, neutral and negative) of the aspect term t in the sentence s.

3.2 DSC Data Preprocessing

In the previous three models using DSC data to help ASC training, there is a problem that is often overlooked, that is, how to label data with sentiment. They usually classify comments according to the number of stars. One star and two-star comments are divided into negative, four-star and five-star comments are regarded as positive, and three-star comments are regarded as neutral. Then label each text according to this classification method and send it to training. All this seems logical, but there is a crucial question here, that is, whether three-star comments should be regarded as neutral comments. In fact, the neutral comment in ASC task means that the comment does not express any sentiment towards the target word. For example, the sentence "*I had a meal in this restaurant at 3 p.m.*" is neutral to the target word "*restaurant*", because it does not make any evaluation on the restaurant. However, this is not the case for neutral comments in DSC tasks. Usually, commentators think something is good, but it is not so good or so bad. For example, we find that many commentators use "*good*" in three-star

comments and "*great*" in five-star comments. Obviously, we cannot regard "*good*" as a neutral evaluation.

Therefore, strictly speaking, there is no neutral comment in DSC comments for ASC tasks. To this end, we give up using three-star comments and only use other star comments. We divide one-star and two-star comments into negative comments, while four-star and five-star comments into positive comments.

Further, we found that both the four-star comments and the five-star comments are actually very satisfied with all aspects of the product, and there are few negative parts in these comments. At the same time, both one-star comments and two-star comments are actually dissatisfied with all aspects of the product, and there are few positive comments in these comments. This is completely different from ASC. Many comments in ASC often have both positive and negative parts. For example, the sentence "*food is delicious, but service is terrible*" gives a positive evaluation of "*food*" and a negative evaluation of "*service*".

In order to make the distribution of DSC data more in line with the characteristics of ASC data, we adopt the method of sentence splicing. Specifically, a positive comment and a negative comment are randomly selected from DSC comments, and they are spliced into a sentence. When the positive comments are in the front and the negative comments are in the back, the spliced sentences are labeled with [0.5, - 0.5], while when the negative comments are in the front and the positive comments are in the back, the spliced sentences are labeled with [-0.5, 0.5]. In addition, positive comments are labeled 1, while negative comments are labeled -1.

Through the above methods, the number of DSC data is increased, and its distribution is more in line with the needs of ASC tasks.

3.3 Pre-trained DSC Module

Before transferring attention knowledge, we first pre-train a DSC module on the large-scale preprocessed DSC datasets. In this work, we employ a conventional attention-based BiLSTM as our DSC module.

For a review document $d = \{w_1^d, w_2^d, \dots, w_n^d\}$, we map it into the corresponding word representations $\{\mathrm{w}_1^d, \mathrm{w}_2^d, \dots, \mathrm{w}_n^d\}$ by looking up an embedding table $E_{emb} \in R^{|v| \times d_e}$, where $|v|$ is the vocabulary size and d_e denotes the word embedding dimension. Then we use a BiLSTM network to obtain the contextual information for each word and generate a sequence of hidden states $\{\mathrm{h}_1^d, \mathrm{h}_2^d, \dots, \mathrm{h}_n^d\}$. Usually, the attention mechanism is used to get the expression of the sentence by aggregating the word contextual representations.

But we didn't do this because we wanted not only the attention of each word, but also the sentiment polarity of each word.

Therefore, next, we directly use a common classification layer to get the sentiment polarity of each word, and use a vector q to calculate the attention of each word where q is a randomly initialized trainable parameter vector. Its significance is to help get the attention of each word.

It is worth noting that the word sentiment here is a scalar, and its value is obtained as follows:

$$w_i^s = (\sigma(h_i^d) - 0.5) * 2 \quad , \quad i = 1, \dots, n \quad (1)$$

Where w_i^s is the word sentiment polarity of h_i^d, σ denotes sigmoid function. It can also be seen from the formula, $w_i^s \in [-1, 1]$.

The word attention here is obtained as follows:

$$w_i^a = \frac{exp(f(h_i^d, q))}{\Sigma_{j=1}^n exp(f(h_j^d, q))}, \quad i = 1, \dots, n \quad (2)$$

$$f(h_i^d, q) = h_i^d \cdot q + b_d, \quad i = 1, \dots, n \quad (3)$$

Where q denotes the query vector, which is a trainable parameter vector. b_d denotes bias.

Finally, we get the sentiment of the whole sentence in the following ways:

$$d_s = \sum_{i=1}^n w_i^a w_i^s \quad (4)$$

In addition, as mentioned earlier, DSC comments have no neutral data. But at the same time, in each sentence, in fact, there are few words with sentiment polarity, and most words do not actually have sentiment polarity, that is, neutral in the common sense. Therefore, we add a regular term to force the emotion of most words close to 0.

The regularizer term is as follows:

$$r_d = \sum_{i=1}^n w_i^s \quad (5)$$

The final loss function is as follows:

$$loss_d = \frac{1}{|D|} \sum_{d \in D} (d_s - y)^2 + \lambda {r_d}^2 \quad (6)$$

where $|D|$ is the size of document data after DSC data preprocessing. λ is hyper-parameter. y denotes the label of DSC data preprocessing.

Please note that, For a comment spliced by a positive comment and a negative comment, if its label is [0.5, - 0.5], it means that the output of the positive part of the first half should be close to 0.5, and the output of the negative part of the second half should be close to - 0.5, and vice versa. The value of 0.5 here shows that we force the whole sentence to pay attention to different parts of the sentence at the same time. In other words, we force the attention distribution of the whole sentence to pay attention to words with different sentiment polarities in a sentence at the same time.

3.4 Attention Transfer Module

The attention obtained through the DSC module focuses on the sentiment words in the whole sentence, while the ASC task only focuses on those sentiment words related to aspect term, so it cannot be used directly and needs to be changed. We use our proposed syntax distance decay method to convert DSC attention into ASC task attention.

First of all, let me define the syntax distance. A sentence is built into a dependency tree, also known as a dependency graph, according to the dependency of its words. Each word is also a node in the dependency graph. The shortest distance between two word nodes in the dependency graph is the syntax distance between the two words.

For any word, we calculate the syntax distance from it to the aspect term. If the aspect term includes multiple words, we calculate and take the shortest distance as the syntax distance from the word to the aspect.

Then we use the following methods to process the word attention obtained by DSC:

$$w_i^{a\prime} = (1 - \frac{l_i}{max(l_i)})w_i^a \qquad (7)$$

Where l_i means the syntax distance from the i-th word to the aspect and $max(l_i)$ means the maximum value of all words' syntax distance.

In order to make the sum of attention after syntax distance decay processing still equal to 1, we further do the following processing.

$$\beta_i = \frac{w_i^{a\prime}}{\sum_{i=1}^{n} w_i^{a\prime}} \qquad (8)$$

Through the above processing, we can get a more attention distribution that meets the needs of ASC.

3.5 ASC Module

As shown on the right side of Figure 1, the architecture of the basic ASC module is similar to that of the DSC module. The difference lies in two points. One is that the ASC task needs to model the aspect term. The other is that we use the traditional method in the ASC model, that is, we obtain the context aware representation of aspect term through the attention mechanism, and then use it for classification. The classification here is still the traditional one hot method, rather than the scalar used in the previous DSC module.

Specifically, given a sentence $s = \{w_1, w_2, \dots, w_n\}$, and an aspect term $t = \{w_l, w_{l+1}, \dots, w_r\}$ in s, we first map each word w_i into its word embedding w_i by looking up the word embedding table $E_{emb} \in R^{|v| \times d_e}$, where $|v|$ is the vocabulary size and d_e denotes the word embedding dimension. Secondly, we send word embedding of each word to BiLSTM to get contextual representation of each word, and we make an average pooling to all words contained in aspect term to obtain the representation of aspect term. Thirdly, we use the attention mechanism to obtain attention of each word towards the aspect term as follows:

$$\alpha_i = \frac{exp(f(h_i,t))}{\Sigma_{j=1}^{n} exp(f(h_i,t))}, \quad i = 1, \dots, n \qquad (9)$$

$$f(h_i, t) = h_i \cdot W_s + b_s, \quad i = 1, \dots, n \qquad (10)$$

Where t denotes the he representation of aspect term. W_s denotes the weight matrix and b_s denotes bias. α_i denotes the word attention obtained from ASC model.

Now we have the processed DSC attention β_i and ASC attention α_i. We use the following methods to fuse the two attentions to obtain the final attention distribution:

$$\gamma_{\mathrm{i}} = (1 - g) \cdot \alpha_i + g \cdot \beta_i, \quad i = 1, \dots, n \qquad (11)$$

Where γ_{i} means the fused word attention of the i-th word. g is a hyper-parameter, which controls the intensity of attention transfer.

Then, the aspect-aware sentence representation r_t is obtained by aggregating the word contextual representations. And through a classifier, we finally get the aspect-level sentiment classification results:

$$r_t = \sum_{i=1}^{n} \gamma_{\mathrm{i}} h_i \qquad (12)$$

$$\hat{y}_i = softmax(W_o r_t + b_o) \qquad (13)$$

$$loss_a = -\sum_{i \in A} y_i \log(\hat{y}_i) \qquad (14)$$

Where $\hat{y}_i$ and y_i respectively are the predictive class distribution and golden class distribution. W_o denotes the weight matrix and b_o denotes bias.

4 Experiments

4.1 Datasets and Settings

Datasets We evaluate our model on two ASC benchmark datasets from SemEval 2014 Task 4 (Pontiki et al.,2014). They respectively contain reviews from Restaurant and Laptop domains. Following previous studies (Tang et al., 2016b; Chen et al., 2017; He et al., 2018), we remove samples with conflicting polarities in all datasets. The statistics of the ASC datasets are shown in Table 1.

Previous models that used DSC data to assist ASC training generally believed that DSC data in similar fields would be helpful to assist ASC data. They generally used the DSC model trained by Amazon comments data to assist the aspect sentiment classification of laptop comments and Yelp comment to assist the aspect sentiment classification of restaurant comments.

Different from them, we only use Yelp comments as the training data of DSC model. Because on the one hand, we think that even in different fields, the expression of sentiment should be close. On the other hand, we pay more attention to data quality than similar fields. After our observation, we found that many sentences in the Amazon dataset are not complete sentences, but part of the original comments, and then labeled with the same label as the original sentence. We believe that if we randomly select a part from a positive sentence, we can not guarantee that the extracted part is still positive, it is likely to be neutral, because as we said before, neutral words are the main body of the sentence, and only a few words have positive or negative sentiment. Because of this, we abandoned Amazon dataset and only used Yelp dataset.

Settings In our experiments, word embeddings are initialized by 300-dimension GloVe (Pennington et al., 2014). All the weight matrices and biases are given the initial value by sampling from the uniform distribution U(-0.1, 0.1).

The dimension of LSTM cell hidden states is set to 200. We employ stochastic gradient descent (SGD) with momentum (Qian, 1999) to train models. The initial learning rate and momentum

Dataset	Pos	Neg	Neu
rRestaurant(Train)	2164	807	637
Restaurant(Test)	728	196	196
Laptop(Train)	994	870	464
Laptop(Test)	341	128	169
Yelp Review	266k	177k	**N/A**

Table 1: Statistics of the datasets.

parameter are respectively set to 0.0003 and 0.9. In addition, we apply dropout (Hinton et al., 2012) with probability 0.5. The hyper-parameter λ and g are respectively set to 1 and 0.5. Finally, we run each model five times and report the average result of them.

4.2 Compared Methods

To demonstrate the superiority of our HITN for ASC tasks, we compare it with followings baselines: ATAE-LSTM (Wang et al., 2016b), PBAN (Gu et al., 2018), PRET+MULT (He et al., 2018) and TransCap (Chen and Qian, 2019), ATN (Chen et al., 2020), DGEDT (Tang et al., 2020), RGAT (Wang et al., 2020).

.

4.3 Main Results

It can be clearly seen from table 2 that our model is superior to the previous three models that use DSC data to assist in training ASC. Even r-gat and dgedt, which have relatively complex comparative structure and excellent performance last year, also have certain advantages. These results prove the effectiveness of our model.

Model	Acc(Restaurant)	F1(Restaurant)	Acc(Laptop)	F1(Laptop)
ATAE-LSTM	78.38	66.36	69.12	63.24
PBAN	78.62	67.45	71.98	66.91
PRET+MULT	78.73	68.63	71.91	68.79
TransCap	79.55	71.41	73.87	70.10
ATN	82.36	74.00	76.48	72.60
DGEDT	**83.90**	75.10	76.80	72.30
R-GAT	83.30	**76.08**	77.42	73.76
HITN(ours)	82.59	74.21	**78.50**	**74.92**

Table 2: Main experiment results.

.

References

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 Task 4: Aspect Based Sentiment Analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, Dublin, Ireland, August. Association for Computational Linguistics.

Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. 2011. Target-dependent Twitter Sentiment Classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 151–160, Portland, Oregon, USA, June. Association for Computational Linguistics.

Svetlana Kiritchenko, Xiaodan Zhu, Colin Cherry, and Saif Mohammad. 2014. NRC-Canada-2014: Detecting Aspects and Sentiment in Customer Reviews. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 437–442, Dublin, Ireland, August. Association for Computational Linguistics.

Duyu Tang, Bing Qin, and Ting Liu. 2016. Aspect Level Sentiment Classification with Deep Memory Network. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 214–224, Austin, Texas, November. Association for Computational Linguistics.

Peng Chen, Zhongqian Sun, Lidong Bing, and Wei Yang. 2017. Recurrent attention network on memory for aspect sentiment analysis. In *Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2018. Exploiting document knowledge for aspect-level sentiment classification. *arXiv preprint arXiv:1806.04346*.

features. In Twenty-Fourth International Joint Conference on Artificial Intelligence.

Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. 2014. Adaptive recursive neural network for target-dependent twitter sentiment classification. In *Annual Meeting of the Association for Computational Linguistics (ACL 2014)*.

Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. 2016a. Recursive neural conditional random fields for aspect-based sentiment analysis. In *Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*.

Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. 2017. Interactive attention networks for aspect-level sentiment classification. In *International Joint Conference on Artificial Intelligence (IJCAI 2017)*.

Xin Li, Lidong Bing, Wai Lam, and Bei Shi. 2018.Transformation networks for target-oriented sentiment classification. In *Annual Meeting of the Association for Computational Linguistics (ACL 2018)*

Yukun Ma, Haiyun Peng, and Erik Cambria. 2018.Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive

Chen Zhang, Qiuchi Li, and Dawei Song. 2019. Aspect-based sentiment classification with aspect-specific graph convolutional networks. *arXiv preprint arXiv:1909.03477*.

Kai Wang, Weizhou Shen, Yunyi Yang, Xiaojun Quan, and Rui Wang. 2020. Relational Graph Attention Network for Aspect-based Sentiment Analysis. *arXiv:2004.12362 [cs]*, April. arXiv: 2004.12362.

Hao Tang, Donghong Ji, Chenliang Li, and Qiji Zhou. 2020. Dependency Graph Enhanced Dual-transformer Structure for Aspect-based Sentiment Classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6578–6588, Online. Association for Computational Linguistics.

Rohan Kumar Yadav, Lei Jiao, Ole-Christoffer Granmo, and Morten Goodwin. 2021. Human-level interpretable learning for aspect-based sentiment analysis. *In The Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21). AAAI.*

Zhengyan Li, Yicheng Zou, Chong Zhang, Qi Zhang, and Zhongyu Wei. 2021. Learning Implicit Sentiment in Aspect-based Sentiment Analysis with Supervised Contrastive Pre-Training. *In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing,* pages 246–256, Online and Punta Cana, Dominican Republic, November. Association for Computational Linguistics.

Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016b. Attention-based LSTM for aspectlevel sentiment classification. *In Conference on Empirical Methods in Natural Language Processing (EMNLP 2016).*

Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018. Learning to attend via word-aspect associative fusion for aspect-based sentiment analysis. In *AAAI Conference on Artificial Intelligence (AAAI 2018)*.

Wei Xue and Tao Li. 2018. Aspect based sentiment analysis with gated convolutional networks. In *Annual Meeting of the Association for Computational Linguistics (ACL 2018)*.