Improving Aspect-based Sentiment Analysis with Gated Graph Convolutional Networks and Syntax-based Regulation

Amir Pouran Ben Veyseh[1]*, **Nasim Nouri***, **Franck Dernoncourt**[2], **Quan Hung Tran**[2], **Dejing Dou**[1] and **Thien Huu Nguyen**[1,3]
[1] Department of Computer and Information Science, University of Oregon, Eugene, OR 97403, USA
[2] Adobe Research, San Jose, CA, USA
[3] VinAI Research, Vietnam
{apouranb,dou,thien}@cs.uoregon.edu,
nasim.nourii@gmail.com, {dernonco,qtran}@adobe.com

Abstract

Aspect-based Sentiment Analysis (ABSA) seeks to predict the sentiment polarity of a sentence toward a specific aspect. Recently, it has been shown that dependency trees can be integrated into deep learning models to produce the state-of-the-art performance for ABSA. However, these models tend to compute the hidden/representation vectors without considering the aspect terms and fail to benefit from the overall contextual importance scores of the words that can be obtained from the dependency tree for ABSA. In this work, we propose a novel graph-based deep learning model to overcome these two issues of the prior work on ABSA. In our model, gate vectors are generated from the representation vectors of the aspect terms to customize the hidden vectors of the graph-based models toward the aspect terms. In addition, we propose a mechanism to obtain the importance scores for each word in the sentences based on the dependency trees that are then injected into the model to improve the representation vectors for ABSA. The proposed model achieves the state-of-the-art performance on three benchmark datasets.

1 Introduction

Aspect-based Sentiment Analysis (ABSA) is a fine-grained version of sentiment analysis (SA) that aims to find the sentiment polarity of the input sentences toward a given aspect. We focus on the term-based aspects for ABSA where the aspects correspond to some terms (i.e., sequences of words) in the input sentence. For instance, an ABSA system should be able to return the negative sentiment for input sentence "*The staff were very polite, but the quality of the food was terrible.*" assuming "*food*" as the aspect term.

Due to its important applications (e.g., for opinion mining), ABSA has been studied extensively in the literature. In these studies, deep learning has been employed to produce the state-of-the-art performance for this problem (Wagner et al., 2016; Dehong et al., 2017). Recently, in order to further improve the performance, the syntactic dependency trees have been integrated into the deep learning models (Huang and Carley, 2019; Zhang et al., 2019) for ABSA (called the graph-based deep learning models). Among others, dependency trees help to directly link the aspect term to the syntactically related words in the sentence, thus facilitating the graph convolutional neural networks (GCN) (Kipf and Welling, 2017) to enrich the representation vectors for the aspect terms.

However, there are at least two major issues in these graph-based models that should be addressed to boost the performance. First, the representation vectors for the words in different layers of the current graph-based models for ABSA are not customized for the aspect terms. This might lead to suboptimal representation vectors where the irrelevant information for ABSA might be retained and affect the model's performance. Ideally, we expect that the representation vectors in the deep learning models for ABSA should mainly involve the related information for the aspect terms, the most important words in the sentences. Consequently, in this work, we propose to regulate the hidden vectors of the graph-based models for ABSA using the information from the aspect terms, thereby filtering the irrelevant information for the terms and customizing the representation vectors for ABSA. In particular, we compute a gate vector for each layer of the graph-based model for ABSA leveraging the representation vectors of the aspect terms. This layer-wise gate vector would be then applied over the hidden vectors of the current layer to produce customized hidden vectors for ABSA. In addition, we propose a novel mechanism to explicitly increase the contextual distinction among the gates

*Equal contribution.

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 4543–4548

to further improve the representation vectors.

The second limitation of the current graph-based deep learning models is the failure to explicitly exploit the overall importance of the words in the sentences that can be estimated from the dependency trees for the ABSA problem. In particular, a motivation of the graph-based models for ABSA is that the neighbor words of the aspect terms in the dependency trees would be more important for the sentiment of the terms than the other words in the sentence. The current graph-based models would then just focus on those syntactic neighbor words to induce the representations for the aspect terms. However, based on this idea of important words, we can also assign a score for each word in the sentences that explicitly quantify its importance/contribution for the sentiment prediction of the aspect terms. In this work, we hypothesize that these overall importance scores from the dependency trees might also provide useful knowledge to improve the representation vectors of the graph-based models for ABSA. Consequently, we propose to inject the knowledge from these syntax-based importance scores into the graph-based models for ABSA via the consistency with the model-based importance scores. In particular, using the representation vectors from the graph-based models, we compute a second score for each word in the sentences to reflect the model's perspective on the importance of the word for the sentiment of the aspect terms. The syntax-based importance scores are then employed to supervise the model-based importance scores, serving as a method to introduce the syntactic information into the model. In order to compute the model-based importance scores, we exploit the intuition that a word would be more important for ABSA if it is more similar the overall representation vector to predict the sentiment for the sentence in the final step of the model. In the experiments, we demonstrate the effectiveness of the proposed model with the state-of-the-art performance on three benchmark datasets for ABSA. In summary, our contributions include:

• A novel method to regulate the GCN-based representation vectors of the words using the given aspect term for ABSA.

• A novel method to encourage the consistency between the syntax-based and model-based importance scores of the words based on the given aspect term.

• Extensive experiments on three benchmark datasets for ABSA, resulting in new state-of-the-art performance for all the datasets.

2 Related Work

Sentiment analysis has been studied under different settings in the literature (e.g., sentence-level, aspect-level, cross-domain) (Wang et al., 2019; Zhang and Zhang, 2019; Sun et al., 2019; Chauhan et al., 2019; Hu et al., 2019). For ABSA, the early works have performed feature engineering to produce useful features for the statistical classification models (e.g., SVM) (Wagner et al., 2014). Recently, deep learning models have superseded the feature based models due to their ability to automatically learn effective features from data (Wagner et al., 2016; Johnson and Zhang, 2015; Tang et al., 2016). The typical network architectures for ABSA in the literature involve convolutional neural networks (CNN) (Johnson and Zhang, 2015), recurrent neural networks (RNN) (Wagner et al., 2016), memory networks (Tang et al., 2016), attention (Luong et al., 2015) and gating mechanisms (He et al., 2018). The current state-of-the-art deep learning models for ABSA feature the graph-based models where the dependency trees are leveraged to improve the performance. (Huang and Carley, 2019; Zhang et al., 2019; Hou et al., 2019). However, to the best of our knowledge, none of these works has used the information from the aspect term to filter the graph-based hidden vectors and exploited importance scores for words from dependency trees as we do in this work.

3 Model

The task of ABSA can be formalized as follows: Given a sentence $X = [x_1, x_2, \ldots, x_n]$ of n words/tokens and the index t ($1 \leq t \leq n$) for the aspect term x_t, the goal is to predict the sentiment polarity y^* toward the aspect term x_t for X. Our model for ABSA in this work consists of three major components: (i) Representation Learning, (ii) Graph Convolution and Regulation, and (iii) Syntax and Model Consistency.

(i) Representation Learning: Following the recent work in ABSA (Huang and Carley, 2019; Song et al., 2019), we first utilize the contextualized word embeddings BERT (Devlin et al., 2019) to obtain the representation vectors for the words in X. In particular, we first generate a sequence of words of the form $\hat{X} = [CLS] + X + [SEP] + x_t + [SEP]$ where $[CLS]$ and $[SEP]$ are the special tokens

in BERT. This word sequence is then fed into the pre-trained BERT model to obtain the hidden vectors in the last layer. Afterwards, we obtain the embedding vector e_i for each word $x_i \in X$ by averaging the hidden vectors of x_i's sub-word units (i.e., wordpiece). As the result, the input sentence X will be represented by the vector sequence $E = e_1, e_2, \ldots, e_n$ in our model. Finally, we also employ the hidden vector s for the special token $[CLS]$ in $\hat{X}$ from BERT to encode the overall input sentence X and its aspect term x_t.

(ii) Graph Convolution and Regulation: In order to employ the dependency trees for ABSA, we apply the GCN model (Nguyen and Grishman, 2018; Veyseh et al., 2019) to perform L abstraction layers over the word representation vector sequence E. A hidden vector for a word x_i in the current layer of GCN is obtained by aggregating the hidden vectors of the dependency-based neighbor words of x_i in the previous layer. Formally, let h_i^l ($0 \leq l \leq L$, $1 \leq i \leq n$) be the hidden vector of the word x_i at the l-th layer of GCN. At the beginning, the GCN hidden vector h_i^0 at the zero layer will be set to the word representation vector e_i. Afterwards, h_i^l ($l > 0$) will be computed by: $h_i^l = ReLU(W_l \hat{h}_i^l)$, $\hat{h}_i^l = \Sigma_{j \in N(i)} h_j^{l-1} / |N(i)|$ where $N(i)$ is the set of the neighbor words of x_i in the dependency tree. We omit the biases in the equations for simplicity.

One problem with the GCN hidden vectors h_i^l GCN is that they are computed without being aware of the aspect term x_t. This might retain irrelevant or confusing information in the representation vectors (e.g., a sentence might have two aspect terms with different sentiment polarity). In order to explicitly regulate the hidden vectors in GCN to focus on the provided aspect term x_i, our proposal is to compute a gate g_l for each layer l of GCN using the representation vector e_t of the aspect term: $g_l = \sigma(W_l^g e_t)$. This gate is then applied over the hidden vectors h_i^l of the l-th layer via the element-wise multiplication $\circ$, generating the regulated hidden vector $\bar{h}_i^l$ for h_i^l: $\bar{h}_i^l = g_l \circ h_i^l$.

Ideally, we expect that the hidden vectors of GCN at different layers would capture different levels of contextual information in the sentence. The gate vectors g_t for these layer should thus also exhibit some difference level for contextual information to match those in the GCN hidden vectors. In order to explicitly enforce the gate diversity in the model, our intuition is to ensure that the regulated GCN hidden vectors, once obtained by *applying different gates to the same GCN hidden vectors*, should be distinctive. This allows us to exploit the contextual information in the hidden vectors of GCN to ground the information in the gate vectors for the explicit gate diversity promotion.

In particular, given the l-th layer of GCN, we first obtain an overall representation vector $\bar{h}^l$ for the regulated hidden vectors at the l-th layer using the max-pooled vector: $\bar{h}^l = max\_pool(\bar{h}_1^l, \ldots, \bar{h}_n^l)$. Afterwards, we apply the gate vectors $g^{l'}$ from the other layers ($l' \neq l$) to the GCN hidden vectors h_i^l at the l-th layer, resulting in the regulated hidden vectors $\bar{h}_i^{l,l'} = g^{l'} \circ h_i^l$. For each of these other layers l', we also compute an overall representation vector $\bar{h}^{l,l'}$ with max-pooling: $\bar{h}^{l,l'} = max\_pool(\bar{h}_1^{l,l'}, \ldots, \bar{h}_n^{l,l'})$. Finally, we promote the diversity between the gate vectors g^l by enforcing the distinction between $\bar{h}^l$ and $\bar{h}^{l,l'}$ for $l' \neq l$. This can be done by minimizing the cosine similarity between these vectors, leading to the following regularization term L_{div} to be added to the loss function of the model:

$$\mathcal{L}_{div} = \frac{1}{L(L-1)} \Sigma_{l=1}^{L} \Sigma_{l'=1, l' \neq l}^{L} \bar{h}^l \cdot \bar{h}^{l,l'}$$

.

(iii) Syntax and Model Consistency: As presented in the introduction, we would like to obtain the importance scores of the words based on the dependency tree of X, and then inject these syntax-based scores into the graph-based deep learning model for ABSA to improve the quality of the representation vectors. Motivated by the contextual importance of the neighbor words of the aspect terms for ABSA, we use the negative of the length of the path from x_i to x_t in the dependency tree to represent the syntax-based importance score syn_i for $x_i \in X$. For convenience, we also normalize the scores syn_i with the softmax function.

In order to incorporate syntax-based scores syn_i into the model, we first leverage the hidden vectors in GCN to compute a model-based importance score mod_i for each word $x_i \in X$ (also normalized with softmax). Afterwards, we seek to minimize the KL divergence between the syntax-based scores $syn_1, \ldots, syn_n$ and the model-based scores $mod_1, \ldots, mod_n$ by introducing the following term L_{const} into the overall loss function: $L_{const} = -syn_i \log \frac{syn_i}{mod_i}$. The rationale is to promote the consistency between the syntax-based and model-based importance scores to facilitate the in-

jection of the knowledge in the syntax-based scores into the representation vectors of the model.

For the model-based importance scores, we first obtain an overall representation vector V for the input sentence X to predict the sentiment for x_t. In this work, we compute V using the sentence representation vector s from BERT and the regulated hidden vectors in the last layer of GCN: $V = [s, max\_pool(\hat{h}_1^L, \ldots, \hat{h}_n^L)]$. Based on this overall representation vector V, we consider a word x_i to be more contextually important for ABSA than the others if its regulated GCN hidden vector $\hat{h}_i^L$ in the last GCN layer is more similar to V than those for the other words. The intuition is the GCN hidden vector of a contextually important word for ABSA should be able capture the necessary information to predict the sentiment for x_t, thereby being similar to V that is supposed to encode the overall relevant context information of X to perform sentiment classification. In order to implement this idea, we use the dot product of the transformed vectors for V and $\hat{h}_i^L$ to determine the model-based importance score for x_i in the model: $mod_i = \sigma(W_V V) \cdot \sigma(W_H \hat{h}_i^L)$.

Finally, we feed V into a feed-forward neural network with softmax in the end to estimate the probability distribution $P(.|X, x_t)$ over the sentiments for X and x_t. The negative log-likelihood $L_{pred} = -\log P(y^*|X, x_t)$ is then used as the prediction loss in this work. The overall loss to train the proposed model is then: $\mathcal{L} = \mathcal{L}_{div} + \alpha\mathcal{L}_{const} + \beta\mathcal{L}_{pred}$ where α and β are trade-off parameters.

4 Experiments

Datasets and Parameters: We employ three datasets to evaluate the models in this work. Two datasets, Restaurant and Laptop, are adopted from the SemEval 2014 Task 4 (Pontiki et al., 2014) while the third dataset, MAMS, is introduced in (Jiang et al., 2019). All the three datasets involve three sentiment categories, i.e., positive, neural, and negative. The numbers of examples for different portions of the three datasets are shown in Table 1.

As only the MAMS dataset provides the development data, we fine-tune the model's hyper-parameters on the development data of MAMS and use the same hyper-parameters for the other datasets. The following hyper-parameters are suggested for the proposed model by the fine-tuning process: 200 dimensions for the hidden vectors of the feed forward networks and GCN layers, 2 hidden layers in GCN, the size 32 for the mini-batches, the learning rate of 0.001 for the Adam optimizer, and 1.0 for the trade-off parameters α and β. Finally, we use the cased BERT$_{base}$ model with 768 hidden dimensions in this work.

Dataset	Pos.	Neu.	Neg.
Restaurant-Train	2164	637	807
Restaurant-Test	728	196	196
Laptop-Train	994	464	870
Laptop-Test	341	169	128
MAMS-Train	3380	5042	2764
MAMS-Dev	403	604	325
MAMS-Test	400	607	329

Table 1: Statistics of the datasets

Results: To demonstrate the effectiveness of the proposed method, we compare it with the following baselines: (1) the feature-based model that applies feature engineering and the SVM model (Wagner et al., 2014), (2) the deep learning models based on the sequential order of the words in the sentences, including CNN, LSTM, attention and the gating mechanism (Wagner et al., 2016; Wang et al., 2016; Tang et al., 2016; Huang et al., 2018; Jiang et al., 2019), and (3) the graph-based models that exploit dependency trees to improve the deep learning models for ABSA (Huang and Carley, 2019; Zhang et al., 2019; Hou et al., 2019; Sun et al., 2019; Wang et al., 2020).

Table 2 presents the performance of the models on the test sets of the three benchmark datasets. This table shows that the proposed model outperforms all the baselines over different benchmark datasets. The performance gaps are significant with $p < 0.01$, thereby demonstrating the effectiveness of the proposed model for ABSA.

Ablation Study: There are three major components in the proposed model: (1) the gate vectors g_l to regulate the hidden vectors of GCN (called **Gate**), (2) the gate diversity component L_{div} to promote the distinction between the gates (called **Div.**), and (3) the syntax and model consistency component L_{const} to introduce the knowledge from the syntax-based importance scores (called **Con.**). Table 3 reports the performance on the MAMS development set for the ablation study when the components mentioned in each row are removed from the proposed model. Note that the exclusion of **Gate** would also remove **Div.** due to their dependency. It is clear from the table that all three components are necessary for the proposed model

Model	Rest.		Laptop		MAMS	
	Acc.	F1	Acc.	F1	Acc.	F1
SVM (2014)	80.2	-	70.5	-	-	-
TD-LSTM (2016)	78.0	66.7	68.8	68.4	74.6	-
AT-LSTM (2016)	76.2	-	68.9	-	77.6	-
MemNet (2016)	79.6	69.6	70.6	65.1	64.6	-
AOA-LSTM (2018)	79.9	70.4	72.6	67.5	77.3	-
CapsNet (2019)	80.7	-	-	-	79.8	-
ASGCN (2019)	80.8	72.1	75.5	69.2	-	-
GAT (2019)	81.2	-	74.0	-	-	-
CDT (2019)	82.3	74.02	77.1	72.9	-	-
R-GAT (2020)	83.3	76.0	77.4	73.7	-	-
BERT* (2019)	84.4	-	77.1	-	82.2	-
GAT* (2019)	83.0	-	80.1	-	-	-
AEN-BERT* (2019)	84.4	76.9	79.9	76.3	-	-
SA-GCN* (2019)	85.8	79.7	81.7	78.8	-	-
CapsNet* (2019)	85.9	-	-	-	83.4	-
R-GAT* (2020)	86.6	81.3	78.2	74.0	-	-
The proposed model	**87.2**	**82.5**	**82.8**	**80.2**	**88.2**	**57.1**

Table 2: Accuracy and F1 scores of the models on the test sets. * indicates the models with BERT.

as removing any of them would hurt the model's performance.

Model	Acc.	F1
The proposed model (full)	**87.98**	**57.2**
-Div.	87.33	56.6
-Con.	86.82	56.2
-Div -Con.	86.52	56.1
-Gate	86.25	55.8
-Gate -Con.	86.02	54.3

Table 3: Ablation study on MAMS dev set

Gate Diversity Analysis: In order to enforce the diversity of the gate vectors g_t for different layers of GCN, the proposed model indirectly minimizes the cosine similarities between the regulated hidden vectors of GCN at different layers (i.e., in L_{div}). The regulated hidden vectors are obtained by applying the gate vectors to the hidden vectors of GCN, serving as a method to ground the information in the gates with the contextual information in the input sentences (i.e., via the hidden vectors of GCN) for diversity promotion. In order to demonstrate the effectiveness of such gate-context grounding mechanism for the diversity of the gates, we evaluate a more straightforward baseline where the gate diversity is achieved by directly minimizing the cosine similarities between the gate vectors g_t for different GCN layers. In particular, the diversity loss term L_{div} in this baseline would be: $\mathcal{L}_{div} = \frac{1}{L(L-1)} \Sigma_{l=1}^{L} \Sigma_{l'=1,l' \neq l}^{L} g_l \cdot g_{l'}$. We call this baseline **GateDiv.** for convenience. Table 4 report the performance of GateDiv. and the proposed model on the development dataset of MAMS. As can be seen, the proposed model is significantly better than GateDiv., thereby testifying to the effectiveness of the proposed gate diversity component with information-grounding in this work. We attribute this superiority to the fact that the regulated hidden vectors of GCN provide richer contextual information for the diversity term L_{div} than those with the gate vectors. This offers better grounds to support the gate similarity comparison in L_{div}, leading to the improved performance for the proposed model.

Model	Acc.
The proposed model	**87.98**
GateDiv.	86.13

Table 4: Model performance on the MAMS development set when the diversity term L_{div} is directly computed from the gate vectors.

5 Conclusion

We introduce a new model for ABSA that addresses two limitations of the prior work. It employs the given aspect terms to customize the hidden vectors. It also benefits from the overall dependency-based importance scores of the words. Our extensive experiments on three benchmark datasets empirically demonstrate the effectiveness of the proposed approach, leading to state-of-the-art results on these datasets. The future work involves applying the proposed model to the related tasks for ABSA, e.g., event detection (Nguyen and Grishman, 2015).

Acknowledgement

This research is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA Contract No. 2019-19051600006 under the Better Extraction from Text Towards Enhanced Retrieval (BETTER) Program. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, the Department of Defense, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein. This document does not contain technology or technical data controlled under either the U.S. International Traffic in Arms Regulations or the U.S. Export Administration Regulations.

References

Dushyant Singh Chauhan, Md Shad Akhtar, Asif Ekbal, and Pushpak Bhattacharyya. 2019. Context-aware interactive attention for multi-modal sentiment and emotion analysis. In *EMNLP*.

Ma Dehong, Li Sujian, Zhang Xiaodong, and Wang Houfeng. 2017. Interactive attention networks for aspect-level sentiment classification. In *IJCAI*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2018. Effective attention modeling for aspect-level sentiment classification. In *ACL*.

Xiaochen Hou, Jing Huang, Guangtao Wang, Kevin Huang, Xiaodong He, and Bowen Zhou. 2019. Selective attention based graph convolutional networks for aspect-level sentiment classification. In *arXiv*.

Mengting Hu, Yike Wu, Shiwan Zhao, Honglei Guo, Renhong Cheng, and Zhong Su. 2019. Domain-invariant feature distillation for cross-domain sentiment classification. In *EMNLP*.

Binxuan Huang and Kathleen Carley. 2019. Syntax-aware aspect level sentiment classification with graph attention networks. In *EMNLP*.

Binxuan Huang, Yanglan Ou, and Kathleen M Carley. 2018. Aspect level sentiment classification with attention-over-attention neural networks. In *SBP-BRiMS*.

Qingnan Jiang, Lei Chen, Ruifeng Xu, Xiang Ao, and Min Yang. 2019. A challenge dataset and effective models for aspect-based sentiment analysis. In *ACL*.

Rie Johnson and Tong Zhang. 2015. Semi-supervised convolutional neural networks for text categorization via region embedding. In *NIPS*.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *EMNLP*.

Thien Huu Nguyen and Ralph Grishman. 2015. Event detection and domain adaptation with convolutional neural networks. In *ACL*.

Thien Huu Nguyen and Ralph Grishman. 2018. Graph convolutional networks with argument-aware pooling for event detection. In *AAAI*.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 task 4: Aspect based sentiment analysis. In *SemEval*.

Youwei Song, Jiahai Wang, Tao Jiang, Zhiyue Liu, and Yanghui Rao. 2019. Attentional encoder network for targeted sentiment classification. In *arXiv*.

Kai Sun, Richong Zhang, Samuel Mensah, Yongyi Mao, and Xudong Liu. 2019. Aspect-level sentiment analysis via convolution over dependency tree. In *EMNLP*.

Duyu Tang, Bing Qin, and Ting Liu. 2016. Aspect level sentiment classification with deep memory network. In *EMNLP*.

Amir Pouran Ben Veyseh, Thien Huu Nguyen, and Dejing Dou. 2019. Graph based neural networks for event factuality prediction using syntactic and semantic structures. In *ACL*.

Joachim Wagner, Piyush Arora, Santiago Cortes, Utsab Barman, Dasha Bogdanova, Jennifer Foster, and Lamia Tounsi. 2014. NRC-canada-2014: Detecting aspects and sentiment in customer reviews. In *SemEval*.

Joachim Wagner, Piyush Arora, Santiago Cortes, Utsab Barman, Dasha Bogdanova, Jennifer Foster, and Lamia Tounsi. 2016. Effective LSTMs for target-dependent sentiment classification. In *COLING*.

Jingjing Wang, Changlong Sun, Shoushan Li, Jiancheng Wang, Luo Si, Min Zhang, Xiaozhong Liu, and Guodong Zhou. 2019. Human-like decision making: Document-level aspect sentiment classification via hierarchical reinforcement learning. In *EMNLP*.

Kai Wang, Weizhou Shen, Yunyi Yang, Xiaojun Quan, and Rui Wang. 2020. Relational graph attention network for aspect-based sentiment analysis. In *ACL*.

Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based LSTM for aspect-level sentiment classification. In *EMNLP*.

Chen Zhang, Qiuchi Li, and Dawei Song. 2019. Aspect-based sentiment classification with aspect-specific graph convolutional networks. In *EMNLP*.

Yuan Zhang and Yue Zhang. 2019. Tree communication models for sentiment analysis. In *ACL*.