Graph Attention Network with Memory Fusion for Aspect-level Sentiment Analysis

Li Yuan[†], **Jin Wang**[†, 1], **Liang-Chih Yu**[‡, *, 2] and **Xuejie Zhang**[†, 3]
[†]School of Information Science and Engineering, Yunnan University, Yunnan, P.R. China
[‡]Department of Information Management, Yuan Ze University, Taiwan
[*]Innovation Center for Big Data and Digital Convergence Yuan Ze University, Taiwan
Contact:{wangjin[1], xjzhang[3]}@ynu.edu.cn, lcyu@saturn.yzu.edu.tw[2]

Abstract

Aspect-level sentiment analysis(ASC) predicts each specific aspect term's sentiment polarity in a given text or review. Recent studies used attention-based methods that can effectively improve the performance of aspect-level sentiment analysis. These methods ignored the syntactic relationship between the aspect and its corresponding context words, leading the model to focus on syntactically unrelated words mistakenly. One proposed solution, the graph convolutional network (GCN), cannot completely avoid the problem. While it does incorporate useful information about syntax, it assigns equal weight to all the edges between connected words. It may still incorrectly associate unrelated words to the target aspect through the iterations of graph convolutional propagation. In this study, a graph attention network with memory fusion is proposed to extend GCN's idea by assigning different weights to edges. Syntactic constraints can be imposed to block the graph convolutional propagation of unrelated words. A convolutional layer and a memory fusion were applied to learn and exploit multiword relations and draw different weights of words to improve performance further. Experimental results on five datasets show that the proposed method yields better performance than existing methods. The code of this paper is availabled at https://github.com/YuanLi95/GATT-For-Aspect.

1 Introduction

Aspect-level sentiment classification is a fine-grained subtask in sentiment analysis (Wang et al., 2019; Peng et al., 2020). Given a sentence and an aspect that appears in the sentence, ASC aims to determine the sentiment polarity of that aspect (e.g., negative, neutral, or positive). For example, a review of a restaurant "*The price is reasonable although the service is poor.*" expresses a *positive* sentiment for the *price* aspect, but also conveys a *negative* sentiment for the *service* aspect, as shown in Figure 1. Such a technique is widely used to analyze online posts reviews, mainly from Amazon reviews or Twitter, to help raise the ability to understand consumer needs or experiences with a product, guiding a manufacturer towards product improvement. Aspect-level sentiment classification is much more complicated than sentence-level sentiment classification. ASC task is necessary to identify the parts of the sentence that describe the correspondence between multiple aspects. Traditional methods mostly use shallow machine learning models with hand-crafted features to build sentiment classifiers for the ASC task (Jiang et al., 2011; Wagner et al., 2014).However, the process for manual feature engineering is time-consuming and labor-intensive as well as limited in classification performance

Recently, with the development of deep learning techniques, various attention-based neural models have achieved remarkable success in ASC. (Wang et al., 2016; Ma et al., 2017; Chen et al., 2017; Gu et al., 2018; Tang et al., 2019). However, these methods ignored the syntactic dependence between context words and aspects in a sentence. As a result, the current attention model may inappropriately focus on syntactically unrelated context words. As shown in Figure 1, when predicting the emotional polarity of *price*, the attention mechanism may focus on the word *poor*, which is not related to its syntax.

To address this issue, Zhang et al. (2019) built a graph convolutional network (GCN) over a dependency tree to exploit syntactical information and word dependencies. However, the model assigns equal weight to the edges connected between words so that words may mistakenly associate syntactically unrelated words to the target aspect through iterations of graph convolutional propagation. As indicated in Figure 1, after three iterations, both *reasonable* (yellow lines) and *poor* (red lines) may be

Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, pages 27–36
December 4 - 7, 2020. ©2020 Association for Computational Linguistics

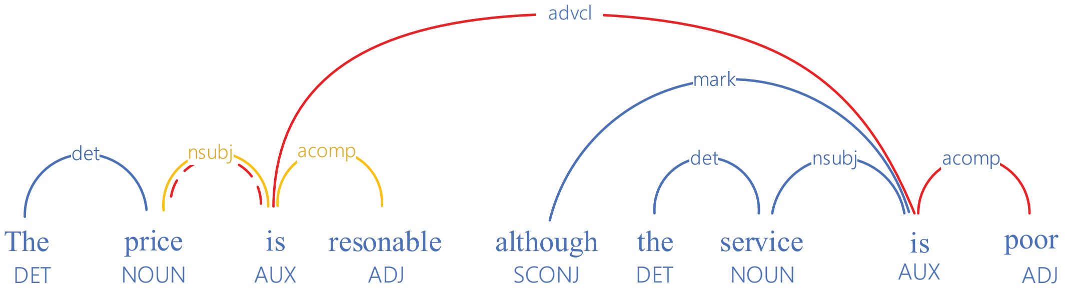


Figure 1: Grammatical Relational Examples.

identified as descriptors of the aspect *price*, which is incorrect. As a result, the model will falsely classify the aspect *price* as a negative sentiment.

In this paper, a graph attention model with memory fusion was proposed. This model extends the idea of graph convolutional networks in two aspects. First, the graph attention mechanism is applied to assign different weights to the edge, so the syntactical constraints can be imposed to block the propagation of syntactically unrelated words to the target aspect. Second, a convolutional operation is applied to extract local information to exploit multiword relations, such as *not good* and *far from perfect*, which can further improve the performance. To integrate all features, a memory fusion layer, which is similar to a memory network, is applied to draw different weights for words according to their contribution to the final classification. Experiments are conducted on five datasets demonstrate how the proposed model outperforms baselines for aspect-level sentiment analysis.

The remainder of this paper is organized as follows. Section 2 briefly reviews the existing works for aspect-level sentiment analysis. Section 3 presents a detailed description of the proposed graph attention model with memory fusion. Section 4 summarizes the implementation details and experimental results. The conclusions of this study are finally drawn in Section 5.

2 Related Works

Aspect-level sentiment classification is an important branch of sentiment classification, aiming to identify the sentiment polarity of an aspect target in a sentence. ASC methods can be divided into traditional and deep learning methods. Traditional methods usually used feature-based machine learning algorithms, such as a feature-based support vector machine (SVM) (Kiritchenko et al., 2014). Due to the inefficiency of manually constructed features, several neural network methods have been proposed for aspect-level sentiment analysis (Jiang et al., 2011), which are mainly based on long short-term memory (LSTM) (Tang et al., 2016a; Wang et al., 2020). Tang et al. (2016b) indicated that the ASC task's challenge is to identify better the semantic correlation between context words and aspect words so that several recent works widely applied an attention mechanism and achieved good performance. Ma et al. (2017) used an interactive attention network to obtain a two-way attention representation of context words and aspect words. Huang et al. (2018) proposed a joint model based on an attention mechanism to model aspects and sentences. Tang et al. (2019) proposed a self-supervised attention model that can dynamically update attention weights.

Yao et al. (2019) introduced the graph convolutional network into the sentiment classification task and achieved good performance. Subsequently, Zhang et al. (2019) proposed to use GCN on the dependency tree of a sentence to exploit the long-range syntactic information for the ASC task.

3 Graph Attention Network with Memory Fusion

The proposed graph attention network with memory fusion is mainly composed of the following four parts: a context encoder, a graph attention layer, a convolutional layer and a memory fusion layer, as shown in Figure 2 .The context encoder employs a vanilla bidirectional LSTM to capture the textual features. It contains a word embedding layer and a BiLSTM layer to produce a hidden representation of the text. Taking the hidden representation as input, the graph attention layer (G-ATT)

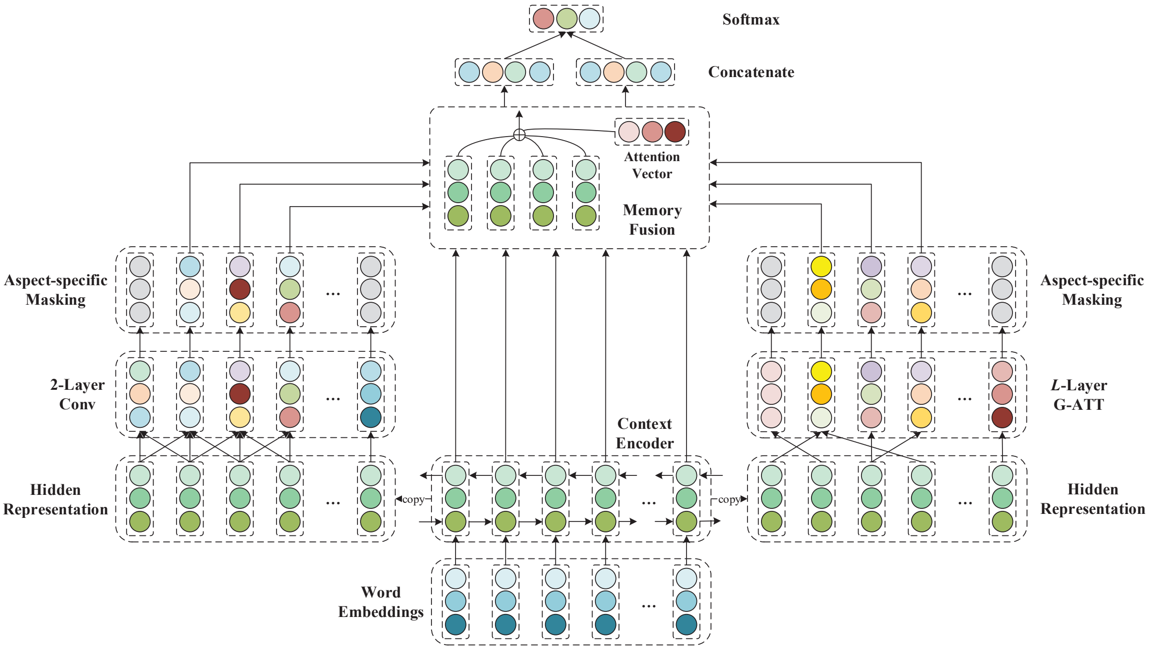


Figure 2: The overall architecture of the proposed graph attention network with memory fusion.

is trained on the dependency tree to mine explicit structural information between words. The convolutional layer was used to extract the local information around the sentiment word, which can dynamically deal with non-single word aspects such as *not good* and *far from perfect*, instead of only taking the average of its vectors. To merge all features, we adopt a memory fusion layer similar to a memory network (Tang et al., 2016b), which can assign different weights to the context words according to their contribution to the final classification. The detailed description is presented as follows.

3.1 Context Encoder

Given a sentence $\mathbf{x}=[x_1, x_2, \cdots, x_{\tau+1}, \cdots, x_{\tau+m}, \cdots x_n]$ containing *n* words, the target aspect starts from the $(\tau+1)$-th word with a length of *m*. A BiLSTM was applied as context encoder, which can capture long-distance dependencies within the sentence. We average the hidden representation of both the forward direction and backward direction to obtain the contextual representation, defined as,

$$(\vec{h}_i^E, \vec{c}_i^E) = LSTM(x_i, \vec{h}_{i-1}^E, \vec{c}_{i-1}^E) \qquad (1)$$

$$(\overleftarrow{h}_i^E, \overleftarrow{c}_i^E) = LSTM(x_i, \overleftarrow{h}_{i+1}^E, \overleftarrow{c}_{i+1}^E) \qquad (2)$$

$$h_i = (\overrightarrow{h_i} \oplus \overleftarrow{h_i})/2 \qquad (3)$$

where $\oplus$ is an element-wise addition operator; $\overrightarrow{h_i} \in \mathbb{R}^{d_h}$, $\overleftarrow{h_i} \in \mathbb{R}^{d_h}$ and $h_i \in \mathbb{R}^{d_h}$ are the forward, backward and output representation, respectively; and d_h is the dimension of hidden state. Thus, the final representation of the context encoder can be denoted as $H^E = [h_1^E, h_2^E, \cdots, h_{\tau+1}^E, \cdots, h_{\tau+m}^E, \cdots, h_{\tau+m}^E]$.

3.2 Graph Attention Layer

The graph attention (G-ATT) layer learns syntactically relevant words to the target aspect on the dependency tree[1], which is widely used in several NLP tasks to effectively identify the relationships and roles of words. After parsing the given sentence as a dependency tree, the adjacency matrix was built from the tree topology. It is worth noting that the dependency tree is a directed graph. Therefore, the graph attention mechanism was applied with consideration of the direction, but the mechanism could be adapted to the undirection-aware scenario. Therefore, we propose a variant on dependency graphs that are undirectional. The obtained hidden state $H^E \in \mathbb{R}^{n \times d_h}$ was fed into a stacked G-ATT model, which was performed in a multilayer fashion with an *L* graph attention layer.

In practice, the representation in the l-the layer was not immediately fed into the G-ATT layer. To enhance the relevance of the context words to the corresponding aspect, we adopted a position weight function to the representation of word i in layer l,

[1] We use spaCy toolkit: https://spacy.io/.

which is widely used in previous works (Li et al., 2018; Zhang et al., 2019), defined as,

$$q_i = \begin{cases} 1 - \frac{\tau+1-i}{n} & 1 \le i < \tau + 1 \\ 0 & \tau + 1 \le i \le \tau + m \\ 1 - \frac{i-\tau-m}{n} & \tau + m < i \le n \end{cases} \quad (4)$$

$$\hat{h}_i^l = q_i h_i^l \quad (5)$$

where $q_i \in \mathbb{R}$ is the position weight to word i.

In each layer, an attention coefficient α_{ij}^l was applied to measure the importance between word i and word j, defined as,

$$\alpha_{i,j}^l = \frac{\exp\left(\mathrm{LeakyReLU}(\mathbf{a}^T[\mathbf{W}_\alpha^l \hat{h}_i^l || \mathbf{W}_\alpha^l \hat{h}_j^l])\right)}{\sum_{k \in \mathcal{N}_i} \exp\left(\mathrm{LeakyReLU}(\mathbf{a}^T[\mathbf{W}_\alpha^l \hat{h}_i^l || \mathbf{W}_\alpha^l \hat{h}_j^l])\right)} \quad (6)$$

where $\mathcal{N}_i$ is the set of the neighbor of word i and $\mathbf{W}_\alpha^l \in \mathbb{R}^{d_h \times d_h}$ is a shared weight matrix applied to perform linear transformation to each word in order to obtain sufficient express ability of high-level representation. $||$ is the concatenation operator, $\mathbf{a} \in \mathbb{R}^{2d_h}$ is a weight vector, and the leaky rectified linear unit (LeakyReLU) is the non-linearity.

To stabilize the learning process of the graph's attention, we implement K different attention with the same parameter settings, which is similar to the multi-head attention mechanism proposed by Vaswani et al. (2017). Thus, the final representation h_i^{l+1} of word i in layer l+1 can be obtained as,

$$h_i^{l+1} = \mathrm{ReLU}(\frac{1}{K}\sum_{k=1}^{K}\sum_{j \in \mathcal{N}_i} \alpha_{i,j}^{l,k} \mathbf{W}_k^l \hat{h}_j^l) \quad (7)$$

where $\alpha_{i,j}^{l,k}$ is the k-th attention coefficients computed by Eq. (6), $\mathbf{W}_k^l$ is the corresponding weight matrix of k-th attention in l-th GAT layer, and the nonlinear function is ReLU. The final representation of the L-layer G-ATT is denoted as $H^L = [h_1^L, h_2^L, \cdots, h_{\tau+1}^L \cdots, h_{\tau+m}^L, \cdots, h_n^L]$, $h_i^L \in \mathbb{R}^{d_h}$.

3.3 Convolutional Layer

The convolutional layer (Conv) was applied to extract local n-gram information which are composed of multiple sentiment words (e.g, *not good* and *far from perfect*), in order to improve the learning ability of the n-gram features. The hidden representation of context encoder H^E is fed into two convolutional layers. In each layer, we use F convolution filters to learn local n-gram features. In a window of ω words $h_{i:i+\omega-1}$,the filter f-th generates the feature map c_i^f as follows,

$$c_i^f = \mathrm{ReLU}(\mathbf{W}^f \circ h_{i:i+\omega-1}^E + b^f) \quad (8)$$

where $\circ$ is a convolutional operator, $\mathbf{W}^f \in \mathbb{R}^{\omega \times d_h}$ and $b^f \in \mathbb{R}^{d_h}$ respectively denote the weight matrix and bias, ω is the length of the filter, and the non-linearity is ReLU. By concatenating all feature maps, the representation for word i will be $h_i^c = [c_i^1, c_i^2, \cdots, c_i^f, \cdots, c_i^F]$. To ensure that the shape of the output is consistent with the shape of the input in the convolutional layer, we set F to d_h and pad each sentence with zero vectors to the maximum input length in the corpora. Then, we send the feature maps to the second convolutional layer, which has a similar structure, to obtain the final representation of convolutional layer $H^C = [h_1^C, h_2^C, \cdots, h_{\tau+1}^C, \cdots, h_{\tau+m}^C, \cdots h_n^C]$, $h_i^C \in {}^{d_h}$.

3.4 Aspect-Specific Masking

The aspect-specific masking layer aims to learn aspect-specific content for memory fusion and the final classification. Therefore, we mask out the hidden state vectors of the input from the G-ATT and Conv layer, i.e., H^L and H^C. Formally, we set all the vectors of non-aspect words to zero and leave the vectors of the aspect words unchanged, defined as,

$$h_i = \begin{cases} 0 & 1 \le i < \tau+1, \tau + m < i \le n \\ h_i & \tau+1 \le \mathrm{i} \le \tau+m \end{cases} \quad (9)$$

The output vector of the G-ATT layer after the mask operation is $H_{masked}^L = [0, \cdots, h_{\tau+1}^L, \cdots, h_{\tau+m}^L, \cdots, 0]$, which has perceived contexts around the aspect so both syntactical dependencies and the long-range multiword relations can be considered. Similarly, the output representation of the convolutional layer after the mask operation is $H_{mask}^C = [0, \cdots, h_{\tau+1}^C, \cdots, h_{\tau+m}^C, \cdots, 0]$.

3.5 Memory Fusion

Memory fusion aims to learn the final representation related to the meaning of aspect words. The idea is to retrieve significant features that are semantically relevant to the aspect words from the hidden representation by aligning the vectors of both G-ATT and Conv to the hidden vectors. Formally, we calculate the attention score for the i-th word in H^E and j-th word in H^L, defined as,

Dataset		Positive	Neutral	Negative	Total	Max Length	Mean Length
Twitter	Train	1561	3127	1560	6248	43	19
	Test	173	346	173	692	39	19
Lap14	Train	994	464	870	2328	81	21
	Test	341	169	128	638	70	17
Rest14	Train	2164	637	807	3608	77	18
	Test	728	196	196	1120	68	17
Rest15	Train	912	36	256	1204	72	15
	Test	326	34	182	542	61	17
Rest16	Train	1240	69	439	1748	72	16
	Test	469	30	117	616	77	18

Table 1: The summary of datasets

$$e_i = \sum_{j=1}^{n} {h_i^L}^T \mathbf{W}_l h_j^E = \sum_{j=\tau+1}^{\tau+m} {h_i^L}^T \mathbf{W}_l h_j^E \quad (10)$$

where $\mathbf{W}_l \in \mathbb{R}^{d_h \times d_h}$ is a bilinear term that interacts with these two vectors and captures the specific semantic relations. According to Socher et al. (2013), such a tensor operator can be used to model complicated compositions between those vectors. Therefore, the attention score weight and final representation of G-ATT are computed as,

$$\alpha_i = \frac{\exp(e_i)}{\sum_{k=1}^{n} \exp(e_k)} \quad (11)$$

$$\mathbf{s}_g = \sum_{i=1}^{n} \alpha_i h_i^E \quad (12)$$

Accordingly, the final representation of the Conv layer is computed as,

$$r_i = \sum_{j=1}^{n} {h_i^C}^T \mathbf{W}_c h_j^E = \sum_{j=\tau+1}^{\tau+m} {h_i^C}^T \mathbf{W}_c h_j^E \quad (13)$$

$$\beta_i = \frac{\exp(r_i)}{\sum_{k=1}^{n} \exp(r_k)} \quad (14)$$

$$\mathbf{s}_c = \sum_{i=1}^{n} \beta_i h_i^E \quad (15)$$

3.6 Sentiment Classification

After obtaining representation $\mathbf{s}_g$ and $\mathbf{s}_c$, they are fed into a fully connected layer and then a *softmax* layer to generate a probability distribution over the classes,

$$\hat{y} = \mathrm{softmax}(\mathbf{W}_s[\mathbf{s}_g || \mathbf{s}_c] + b_s) \quad (16)$$

where $\mathbf{W}_s$ and b_s respectively denote the weights and bias in the output layer. Thus, given a training set $\{\mathbf{x}^{(t)}, y^{(t)}\}_{t=1}^{T} = 1$, where $\mathbf{x}^{(t)}$ is a training sample, $y^{(t)}$ is the corresponding actual sentiment label, and T is the number of training samples in the corpus. The training goal is to minimize the cross-entropy $\mathcal{L}_{cls}(\theta)$ defined as,

$$\mathcal{L}_{cls}(\theta) = -\frac{1}{T} \sum_{t=1}^{T} \log p(\hat{y}^{(t)} | \mathbf{x}^{(t)}; \theta) + \lambda \, \|\theta\|_2^2 \quad (17)$$

where θ denotes all trainable parameters. To avoid overfitting, an L_2-regularization $\lambda \, \|\theta\|_2^2$ is also introduced to the loss function in the training phase, where λ is the decay factor.

4 Experimental Results

This section conducts comparative experiments on five corpora against several previously proposed methods for aspect-level sentiment analysis. The experimental setting and empirical results are then presented in detail.

4.1 Dataset

To compare the proposed model with other aspect-level sentiment analysis models, we conduct experiments on the following five commonly used datasets: **Twitter** was originally proposed by Dong et al. (2014) and contains several Twitter posts, while the other four corpora (**Lap14**, **Rest14**, **Rest15**, **Rest16**) were respectively retrieved from SemEval 2014 task 4 (Pontiki et al., 2014), SemEval 2015 task 12 (Pontiki et al., 2015) and SemEval 2016 Task 5 (Pontiki et al., 2016), which include two types of data, i.e., reviews of laptops and restaurants. The statistical descriptions of these corpora are shown in Table 1. We use accuracy and Macro-average F_1-score as evaluation metrics; these are commonly used in ASC task (Huang and Carley, 2019; Zhang et al., 2019). A higher accuracy or F_1-score indicates better prediction performance

Model	Twitter		Lap14		Rest14		Rest15		Rest16	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
LSTM	69.56	67.70	69.29	63.09	78.13	67.47	77.37	55.17	86.80	63.88
TD-LSTM	70.81	69.11	70.45	64.78	79.47	69.01	78.23	57.25	87.17	64.89
MemNet	71.48	69.90	70.64	65.17	79.61	69.64	77.31	58.28	85.44	65.99
IAN	72.50	70.81	72.05	67.38	79.26	70.09	78.54	52.65	84.74	55.21
RAM	69.36	67.30	74.49	**71.35**	80.23	70.80	78.85	**61.97**	88.92	68.23
AOA	72.30	70.20	72.62	67.52	79.97	70.42	78.17	57.02	87.50	66.21
TNet-LF	**72.98**	**71.43**	74.61	70.14	80.42	71.03	78.47	59.47	**89.07**	**70.43**
ASGCN	72.15	70.40	**75.55**	71.05	**80.77**	**72.02**	**79.89**	61.89	88.99	67.48
G-ATT-U	73.60	**72.12**	**76.18**	**72.23**	**81.59**	**72.65**	8**1.18**	**64.07**	**89.06**	71.97
G-ATT-D	**73.89**	71.82	75.75	71.52	80.89	71.68	80.93	64.03	88.81	**72.36**

Table 2: Model comparison results (%). In the case of random initialization, the average accuracy of the 3 runs and the macro F_1-score. The best results of its baseline model and our model are shown in bold.

Model	Twitter		Lap14		Rest14		Rest15		Rest16	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
ASGCN-DG	72.15	70.40	75.55	71.05	80.77	72.02	79.89	61.89	88.99	67.48
G-ATT-U	73.60	**72.12**	**76.18**	**72.23**	**81.59**	**72.65**	**81.18**	**64.07**	**89.06**	**71.97**
G-ATT-U w/o Pos	**73.74**	**72.00**	75.13	71.26	**81.82**	**73.91**	80.07	62.42	**88.69**	69.54
G-ATT-U w/o Mask	73.36	71.47	**75.24**	70.70	80.15	70.49	79.89	**62.78**	88.53	**70.34**
G-ATT-U w/o GAT	73.03	71.04	74.56	71.23	80.21	71.16	80.38	61.31	87.66	68.27
G-ATT-U w/o Conv	73.23	71.22	74.82	**71.35**	80.86	71.77	**80.54**	62.02	87.39	69.22

Table 3: Ablation study results (%). Accuracy and macro F1-scores are the average value over 3 runs with random initialization.

4.2 Implementation Details

To comprehensively evaluate the proposed model, we selected the following baseline methods, which are introduced as follows:

- **LSTM** (Tang et al., 2016a) uses the standard LSTM model to send the state of the last layer to the *softmax* layer to obtain the output of sentiment probability.

- **TD-LSTM** (Tang et al., 2016a) connects aspect word embedding and context word embedding to obtain the final word embedding representation, and the two sides of the aspect word are respectively modeled by LSTM to obtain the hidden layer representation.

- **MemNet** (Tang et al., 2016b) consists of a multilevel memory network, which effectively retains context and aspect information.

- **IAN** (Ma et al., 2017) exchanges information between context and aspect as an interactive attention model.

- **RAM** (Chen et al., 2017) learns sentence representation by layers consisting of an attention-based aggregation of word features and a GRU cell with multilayer architecture.

- **AOA** (Huang et al., 2018) captures the interaction between context and aspect words by jointly modeling aspects and sentences.

- **TNet-LFT** (Li et al., 2018) increases the retention of context information through a context retention conversion mechanism.

- **ASGCN** (Zhang et al., 2019) uses external grammatical information through the graph convolution neural network, while aspect obtains syntax-related context information.

- **G-ATT** uses either undirectional (**G-ATT-U**) or directional (**G-ATT-D**) graphs to represent the parsed tree-structure as the proposed model.

For all the models, the 300-dimensional GloVe vector (Pennington et al., 2014) pretrained on 840B Common Crawl was used as the initial word embedding. Words that do not appear in GloVe were initialized with a uniform distribution of U(-0.25, 0.25). The hidden layer vectors' dimensions are all 300, and all model weights are initialized with the *Xavier* normalization (Glorot and Bengio, 2010). RMSprop was used as the optimizer with a learning rate of 0.001 to train all the models. We set the L_2-regularization decay factor to 1e-4 and the batch size to 40. The negative input slope of LeakyReLU in the G-ATT layer is set to 0.2. All aforementioned

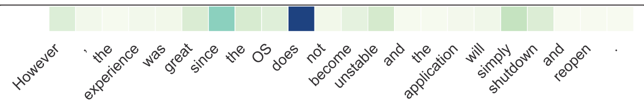
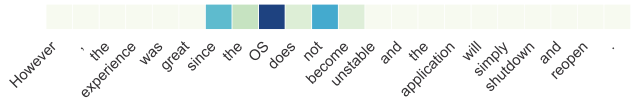
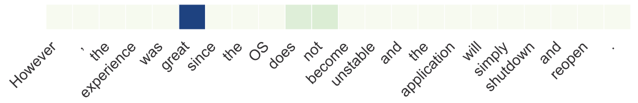
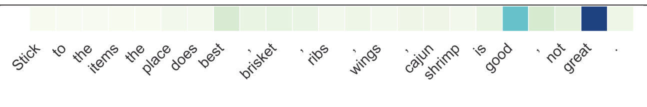
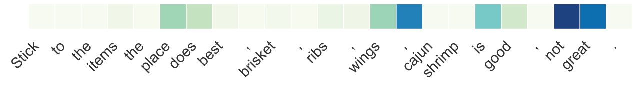
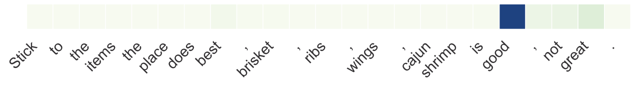
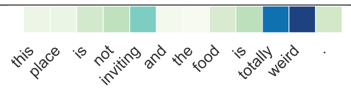
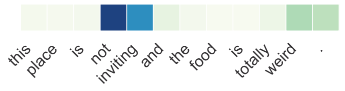
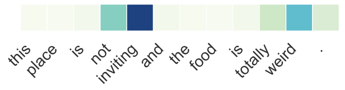
Aspect	Model	Attention Visualization	Prediction	Label
OS	ASGCN		neutral	positive
	Conv		positive	positive
	G-ATT			
Cajun shrimp	ASGCN		negative	positive
	Conv		positive	positive
	G-ATT			
Place	ASGCN		neutral	negative
	Conv		negative	negative
	G-ATT			

Table 4: Visualization of the proposed model.

hyperparameters are selected using a grid-search strategy. The epoch was set depending on an early stop strategy. The training processing stops after five epochs if there is no improvement. The experimental results are obtained by averaging the results of three random initialization runs.

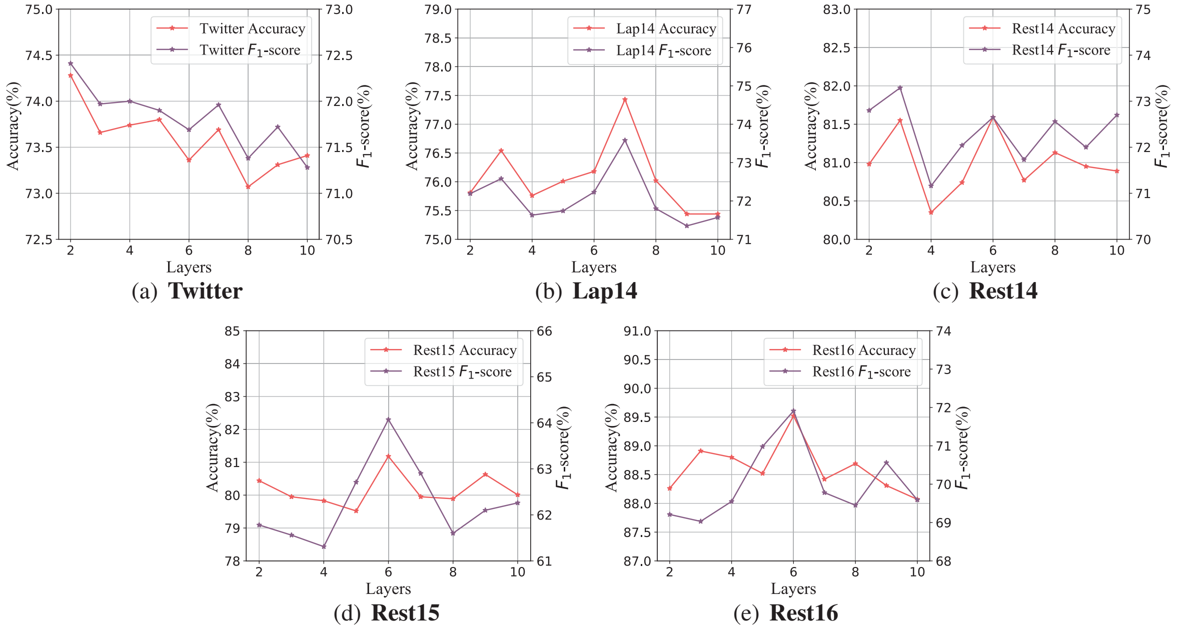
4.3 Comparative Results

Table 2 shows the comparative results of G-ATT-D and G-ATT-U against several baselines. As indicated, G-ATT-U outperformed all baseline models by using F_1-score as a criterion. In terms of accuracy, except results slightly lower than the TNet-LF model on **Rest16**, both G-ATT-D and G-ATT-U achieved better performance. The rational reason is that the proposed model can capture both syntactic and local information, thus improving performance.

In addition, the improvement of the F_1-score of the proposed model on **Rest15** and **Rest16** is huge compared to the baselines, which is 2.1% and 1.5%, respectively. The possible reason is that the syntactical structure of the texts in **Rest15** and **Rest16** is more complicated than those in **Twitter**, **Lap14** and **Rest14**. The performance of directional version (G-ATT-D) is slightly higher than the undirectional version (G-ATT-U) on **Twitter**, **Rest15** and **Rest16**, while performance is slightly lower on **Lap14** and **Rest14**, indicating an undirectional syntax relationship that is more appropriate on those datasets.

4.4 Ablation Experiment

Table 3 shows the ablation experiments to investigate further how the models can benefit from each component. As indicated, removing the position weight (i.e., G-ATT-U w/o Pos) causes the performance on **Lap14**, **Rest15** and **Rest16** to decrease. However, the performance of G-ATT-U w/o Pos increases F_1-score by 1.26% when used on **Twitter** and **Rest14** since the local information is less important than syntactic. Removing the mask op-



(a) **Twitter** (b) **Lap14** (c) **Rest14**

(d) **Rest15** (e) **Rest16**

Figure 3: Effect of the number of G-ATT Layers

eration (i.e., G-ATT-U w/o mask) reduces the performance, which shows that the mask operation prevents the noise word from entering the final representation. Further, **Twitter**, **Lap14**, and **Rest14** are less syntactical, so the integration of position weight does not benefit or can even negatively benefit the results.

Besides, it is observed that G-ATT-U w/o Conv is generally better than G-ATT-U w/o G-ATT, which shows that the GAT layer benefits for the model are greater than the Conv layer, indicating that the contextual syntax-related information is more important than local information. Compared with ASGCN-DG, the proposed G-ATT-U w/o Conv achieved better performance, especially on **Twitter** and **Rest16**, with F_1-score improvements of 0.64% and 1.74%, respectively. This result shows that G-ATT-U w/o Conv outperformed the ASGCN model in most cases, indicating that graph attention layers with different edge weights are more effective than graph convolution layers with equal edge weights.

4.5 Visualization

Memory fusion can capture both syntax-related and local information with the attention mechanism. For visualization, we selected three examples from **Lap14** and **Rest16** that are significantly improved by the proposed G-ATT model against the ASGCN-DG model. We conducted a visualization experiment using a heat map to show the attention score offered by parameters α and β in Eq.(11) and Eq.(14), respectively, as shown in Table 4. The color density is the attention score of each token. A deeper color indicates that more weight is assigned to the token according to its contribution to the final classification. As indicated, ASGCN allows the syntactically unrelated words to be associated with the target aspect by assigning equal weight to the edge, such as *great* for OS, *good* for *Cajun shrimp* and *not inviting* for *place*. Conversely, G-ATT-U tends to block graph convolution propagation from unrelated words to the target aspect by assigned attention weights to the edges. The convolution operation can also exploit some explicit structure, such as *not great* and *not inviting*. Such phrases are expressive and task-specific, thus improve performance.

4.6 Number of GAT layers

Since G-ATT involves L layers of graph attention, we investigate whether the number of layers can determine the proposed model's performance. As indicated, the best performance can be achieved when L is 2 on **Twitter**, 7 on **Lap14**, 3 on **Rest14** and 6 on **Rest15** and **Rest16**. When L is greater than 7, a decreasing trend in both metrics is presented. As L reaches 10, the model contains too many parameters and becomes more difficult to train.

5 Conclusions

In this study, a graph attention network with memory fusion is proposed for aspect-level sentiment analysis. A graph attention layer was implemented to capture a context word's syntactic relationship to the target aspect by learning different weights for edges to block the propagation from unrelated words. Moreover, a convolutional layer and a memory fusion were used to learn the local information and draw different weights for context words. Experimental results show that the G-ATT model yields better performance than the existing methods for aspect-based sentiment analysis. Besides, ablation studies and case studies are provided to prove the effectiveness of the proposed model further. Future works will improve the graph attention layer and dynamic to learn the attention score, so the proposed model can better integrate syntax-related context information.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant No. 61966038, 61702443 and 61762091, and in part by the Ministry of Science and Technology, Taiwan, ROC, under Grant No. MOST107-2628-E-155-002-MY3. The authors would like to thank the anonymous reviewers for their constructive comments.

References

Peng Chen, Zhongqian Sun, Lidong Bing, and Wei Yang. 2017. Recurrent attention network on memory for aspect sentiment analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP-2017)*, pages 452–461.

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 *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL-2014)*, pages 49–54.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256.

Shuqin Gu, Lipeng Zhang, Yuexian Hou, and Yin Song. 2018. A position-aware bidirectional attention network for aspect-level sentiment analysis. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING-2018)*, pages 774–784.

Binxuan Huang and Kathleen M Carley. 2019. Parameterized convolutional neural network for aspect-level sentiment classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP-2018)*, pages 1091–1096.

Binxuan Huang, Yanglan Ou, and Kathleen M. Carley. 2018. Aspect level sentiment classification with attention-over-attention neural networks. In *Proceedings of the International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation*, pages 197–206.

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 (ACL-2011)*, pages 151–160.

Svetlana Kiritchenko, Xiaodan Zhu, Colin Cherry, and Saif M. 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.

Xin Li, Lidong Bing, Wai Lam, and Bei Shi. 2018. Transformation networks for target-oriented sentiment classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL-2018)*, pages 946–956.

Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. 2017. Interactive attention networks for aspect-level sentiment classification. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI-2017)*, pages 4068–4074.

Bo Peng, Jin Wang, and Xuejie Zhang. 2020. Adversarial learning of sentiment word representations for sentiment analysis. *Information Sciences*, 541:426–441.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP-2014)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, Véronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeniy Kotelnikov, Nuria Bel, Salud María Jiménez-Zafra, and Gülşen Eryiğit. 2016. SemEval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 19–30, San Diego, California. Association for Computational Linguistics.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. Semeval-2015 task 12: aspect based sentiment analysis. In *Proceedings ofthe 9th International Workshop on Semantic Evaluation (SemEval-2015)*, pages 486–495.

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, Proceedings (SemEval-2014)*, pages 27–35.

Richard Socher, Alex Perelygin, and Jy Wu. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP-2013)*, pages 1631–1642.

Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. 2016a. Effective LSTMs for target-dependent sentiment classification. In *Proceedings of the 26th International Conference on Computational Linguistics (COLING-2016)*, pages 3298–3307.

Duyu Tang, Bing Qin, and Ting Liu. 2016b. Aspect level sentiment classification with deep memory network. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2016)*, pages 214–224.

Jialong Tang, Ziyao Lu, Jinsong Su, Yubin Ge, and Linfeng Song. 2019. Progressive self-supervised attention learning for aspect-level sentiment analysis. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL-2019)*, pages 557–566. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems (NIPS-2017)*, pages 5999–6009.

Joachim Wagner, Piyush Arora, Santiago Cortes, Utsab Barman, Dasha Bogdanova, Jennifer Foster, and Lamia Tounsi. 2014. DCU: Aspect-based polarity classification for semeval task 4. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval-2014)*, pages 223–229.

Jin Wang, Liang-chih Yu, K Robert Lai, and Xuejie Zhang. 2016. Community-Based Weighted Graph Model for Valence-Arousal Prediction of Affective Words. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(11):1957–1968.

Jin Wang, Liang-Chih Yu, K. Robert Lai, and Xuejie Zhang. 2019. Investigating dynamic routing in tree-structured LSTM for sentiment analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3432–3437. Association for Computational Linguistics.

Jin Wang, Liang Chih Yu, K. Robert Lai, and Xuejie Zhang. 2020. Tree-structured regional CNN-LSTM model for dimensional sentiment analysis. *IEEE/ACM Transactions on Audio Speech and Language Processing*, 28:581–591.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI-2019)*, pages 7370–7377.

Chen Zhang, Qiuchi Li, and Dawei Song. 2019. Aspect-based sentiment classification with aspect-specific graph convolutional networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP-2019)*, pages 4568–4578.