Aspect-Based Sentiment Analysis With Heterogeneous Graph Neural Network

Wenbin An, Feng Tian, *Senior Member, IEEE*, Ping Chen, and Qinghua Zheng, *Member, IEEE*

***Abstract*— Aspect-based sentiment analysis aims to predict sentiment polarities of given aspects in text. Most current approaches employ attention-based neural methods to capture semantic relationships between aspects and words in one sentence. However, these methods ignore the fact that sentences with the same aspect and sentiment polarity often share the structure and semantic information in a domain, which leads to lower model performance. To mitigate this problem, we propose a heterogeneous aspect graph neural network (HAGNN) to learn the structure and semantic knowledge from intersentence relationships. Our model is a heterogeneous graph neural network since it contains three different kinds of nodes: word nodes, aspect nodes, and sentence nodes. These nodes can pass structure and semantic information between each other and update their embeddings to improve the performance of our model. To the best of our knowledge, we are the first to use a heterogeneous graph to capture relationships between sentences and aspects. The experimental results on five public datasets show the effectiveness of our model outperforming some state-of-the-art models.**

***Index Terms*— Aspect-based sentiment analysis (ABSA), aspect-category sentiment analysis (ACSA), heterogeneous graph neural network (GNN).**

I. Introduction

ASPECT-BASED sentiment analysis (ABSA) [1]–[3] can provide more fine-grained information than the traditional sentiment analysis. ABSA contains several subtasks, and one of them is aspect-category sentiment analysis (ACSA) [4]. ACSA aims at predicting sentiment polarities on predefined aspect categories, which may not occur in the sentence. For example, in the sentence "while the restaurant was large and a bit noisy, the drinks were fantastic," ACSA will predict the sentiment polarity for the aspect-category ambience even though it does not appear in the text. This characteristic can easily lead to wrong sentiment predictions because of the diverse expressions toward the same aspect category.

To extract semantic relationships between sentences and concerned aspect categories, most current efforts used neural networks with attention mechanism to guide models to focus on tokens related to the concerned aspect categories [4]–[6]. The advantage of these models was that they can extract aspect-related information from a sentence to generate aspect embeddings and apply an attention mechanism to get the final sentiment polarity based on the sentence and aspect embeddings. However, these models generated aspect and sentence embeddings based on only one sentence and ignored interactions between sentences, which may lose information from intersentence relationships, especially for sentences with the same aspect category and sentiment polarity.

Two types of pattern knowledge can be learned from the interactions between sentences and aspect categories: structure patterns and semantic patterns, which are learned from similar sentence structures and diverse semantic expressions between sentences with the same aspect category and sentiment polarity. First, sentences with the same aspect category and sentiment polarity sometimes share similar structures (e.g., similar sentence structures or word usage). For example, in Table I, sentences $S1$ and $S2$ mention the same aspect categories with the same sentiment polarities, and their sentence structures are similar, so they can learn structure-level knowledge from each other to strengthen their own features. Second, sentences with the same aspect category and sentiment polarity sometimes use different semantic expressions toward the same aspect category (e.g., specific or general and high level or low level) even though their sentence structures are similar. For example, in Table I, sentences $S3$ and $S4$ mention the same aspect category with the same sentiment polarity, and their sentence structures are similar, but their expressions are in different semantic levels (the former is general and the latter is specific), and knowledge learned from diverse semantic expressions can improve the generalizability of our model. Even though there are many methods proposed to extract structure and semantic information from one individual sentence, how to gather structure and semantic information from multiple related sentences to improve model performance is yet to be explored.

To mitigate this gap, we propose a heterogeneous aspect graph neural network (HAGNN), where structure and semantic

Manuscript received June 1, 2021; revised October 24, 2021; accepted January 23, 2022. This work was supported in part by the National Key Research and Development Program of China under Grant 2020AAA0108800; in part by the National Natural Science Foundation of China under Grant 62137002, Grant 61721002, Grant 61937001, Grant 61877048, and Grant 62177038; in part by the Innovation Research Team of Ministry of Education under Grant IRT_17R86; in part by the Project of China Knowledge Centre for Engineering Science and Technology; in part by the Natural Science Basic Research Plan in Shaanxi Province of China under Grant 2020JM-070; and in part by the Ministry of Education-China Mobile Communication Corporation (MoE-CMCC) "Artificial Intelligence" Project under Grant MCM20190701; in part by the Project of Chinese Academy of Engineering (The Online and Offline Mixed Educational Service System for "The Belt and Road" Training in MOOC China). *(Corresponding author: Feng Tian.)*

Wenbin An and Feng Tian are with the School of Automation Science and Engineering, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: wenbinan@stu.xjtu.edu.cn; fengtian@mail.xjtu.edu.cn).

Ping Chen is with the Department of Engineering, University of Massachusetts, Boston, MA 02125 USA (e-mail: ping.chen@umb.edu).

Qinghua Zheng is with the School of Computer Science and Technology, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: qhzheng@mail.xjtu.edu.cn).

Digital Object Identifier 10.1109/TCSS.2022.3148866

TABLE I
TWO EXAMPLES OF RESTAURANT REVIEWS THAT HAVE THE SAME ASPECT CATEGORY AND SENTIMENT POLARITY. THE FIRST EXAMPLE SHOWS SIMILAR STRUCTURE-LEVEL EXPRESSIONS OF SENTENCES. THE SECOND EXAMPLE SHOWS DIFFERENT SEMANTIC-LEVEL EXPRESSIONS OF SENTENCES

Id	Restaurant reviews	Aspect	Sentiment
S1	**The food was mediocre** and **the service was severely slow.**	food service	neutral negative
S2	**The appetizers are ok**, but **the service is slow.**	food service	neutral negative
S3	You're going to go back because **the food was good.**	food	positive
S4	You're going to go back because **the crusted pizza was good.**	food	positive

information can flow between sentences and aspect categories through the graph. HAGNN contains three kinds of nodes named word nodes, sentence nodes, and aspect nodes. One high-level diagram of HAGNN is shown in Fig. 1. The word nodes act as the additional intermediary, which can pass information between sentences according to word co-occurrence. The aspect nodes can aggregate different semantic-level information from sentences with the same aspect category and sentiment polarity. The sentence nodes can get structure information from other similar sentences through the word and aspect nodes. With information passing through the graph, these node embeddings will be iteratively updated with information from other related nodes. Also, the final representations of sentence nodes and aspect nodes will be used to predict the sentiment polarities on given aspect categories.

The main contributions of our work can be summarized as follows.

1) We study the insight that sentences with the same aspect category and sentiment polarity can share similar sentence structures and have various semantic expressions, where common structure and semantic information can be used to improve the model performance.
2) We propose an HAGNN for ACSA, which can pass structure and semantic information between related sentences and aspect categories through HAGNN by updating node embeddings. To the best of our knowledge, we are the first one to use a heterogeneous graph to capture relationships between sentences and aspect categories for ACSA.
3) Our model outperforms the best compared models by 1.25% of average accuracy and 1.16% of average macro-F1 on five public datasets, which demonstrates the effectiveness of our model.

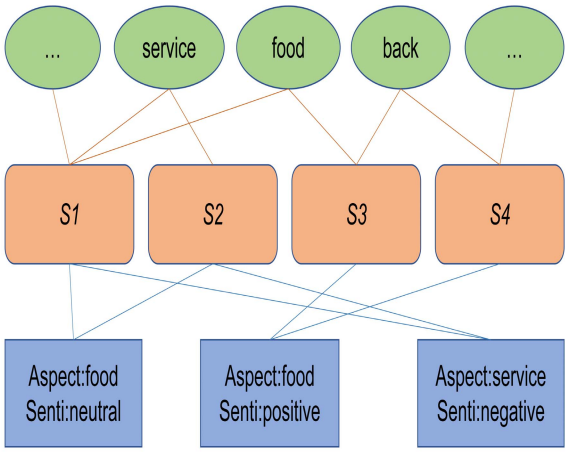

Fig. 1. High-level diagram of our heterogeneous graph for ACSA. *S*1, *S*2, *S*3, and *S*4 are sentences in Table I, and senti represents the sentiment polarity toward an aspect category.

II. RELATED WORK

ACSA aims at predicting sentiment polarities of a sentence toward predefined aspect categories. With the development of deep learning, significant progress has been made in the area of ACSA. Most of the current models are based on the attention mechanism or gating mechanism, which can guide these models to focus on the aspect categories to be analyzed.

In terms of neural network topology, current methods can be mainly divided into four types: recurrent neural network (RNN)-, convolutional neural network (CNN)-, graph neural network (GNN)- and bidirectional encoder representations from transformers (BERT)-based. For RNN-based models, Wang *et al.* [5] first used attention-based long short-term memory (LSTM) networks to generate aspect-specific embeddings and got the final representations by concatenating sentence embeddings and aspect embeddings. Furthermore, Aydin and Güngör [7] combined recurrent and recursive neural models to perform ABSA. For CNN-based models, Xue and Li [4] used CNNs as feature extractors and used gating units to control the information flow, and since these components can be easily parallelized during training, the model was more efficient. For GNN-based models, Li *et al.* [8] proposed a sentence constituent-aware network (SCAN) for ACSA, SCAN contained two graph attention networks (GATs), and an interactive loss function that can perform aspect-category detection task and ACSA task jointly. Tang *et al.* [9] implemented a dependency graph enhanced dual-transformer network for

ACSA, and graph convolutional networks based on semantic dependency tree were used in the model. Wang *et al.* [10] proposed a relational GAT to encode dependency parse tree structure for sentiment predictions. To tackle the problem of relevant syntactical constraints and long-range word dependencies, Zhang *et al.* [6] built a graph convolutional network based on the dependency tree of a sentence and outperformed a range of models. Liu *et al.* [11] proposed a dynamic heterogeneous graph to jointly model the opinion aspects and their sentiment polarities. To represent heterogeneous relations between different events, Zheng *et al.* [12] introduced a heterogeneous-event graph network. Furthermore, Xu *et al.* [13] defined sentence and aspect nodes to learn the sentence and aspect representations separately with a heterogeneous graph convolutional network. For BERT-based models, Sun *et al.* [14] fine-tuned the BERT model and achieved the state-of-the-art results. There are also some other methods. For example, Jiang *et al.* [15] proposed a capsule-guided routing mechanism, in which the prior knowledge about sentiment categories is stored in the capsules. He *et al.* [16] proposed an interactive multitask learning network to perform ACSA with aspect-category detection simultaneously to fully exploit the joint information.

However, existing models just exploited and learned information from one sentence and ignored structure and semantic interactions between sentences, which may lose information. As mentioned in Section I, sentences with the same aspect category and sentiment polarity can have similar structures, and structure similarity information can be shared between sentences to strengthen their features and improve the representation capacity of our model. Meanwhile, there are diverse semantic expressions toward the same aspect category, and semantic diversity information can be used to update representations of aspect categories to improve the generalizability of our model. Thus, how to extract and take advantage of structure and semantic information from related sentences is the main innovation and research contribution of our work.

III. Model

The task of ACSA can be formulated as follows. Given a sentence $S = \{w_1, w_2, \ldots, w_n\}$, where w_i is the ith word in S, and the aspect categories mentioned in the sentence $C^S = \{C_1^S, C_2^S, \ldots, C_K^S\}$, where C_k^S is a predefined aspect category, a model should predict sentiment polarities toward the aspect categories C^S, $P = \{P_1^S, P_2^S, \ldots, P_K^S\}$, where $P_k^S \in$ {Positive, Neutral, Negative}.

To better exploit the structure and semantic relationships between sentences, we propose an HAGNN to pass structure and semantic information between sentences. There are three kinds of nodes in our model: word nodes, sentence nodes, and aspect nodes. Sentence nodes connect with word nodes according to word occurrence and connect with aspect nodes based on aspect categories and sentiment polarities. Word nodes play as intermediary between sentence nodes, sentence nodes can aggregate structure information from other sentence nodes through word and aspect nodes, and aspect nodes can learn diverse semantic expressions from different sentence nodes, which can improve the generalizability of our model. After structure and semantic information passing through HAGNN, representations of sentence nodes and aspect nodes will be used to predict sentiment polarities.

To learn more fine-grained aspect representations, we split one aspect node into M nodes in our model, where M is the number of sentiment polarities, and the aspect node V_a^{km} means the kth aspect category with the mth sentiment polarity. In this way, we can learn more fine-grained aspect representations with specific sentiment information than traditional methods that all sentiment polarities share the same representations. When predicting, sentence nodes will compute similarities with aspect nodes and decide the final sentiment polarity toward the given aspect category.

A. Heterogeneous Graph for ACSA

The heterogeneous graph in our model can be formulated as $\mathrm{G} = \{V_w, V_s, V_a, E_{ws}, E_{sa}\}$, where V_w, V_s, and V_a denote the word nodes, sentence nodes, and aspect nodes, respectively; E_{ws} indicates the edges between word nodes and sentence nodes; and E_{sa} are the edges between sentence nodes and aspect nodes.

Fig. 2 shows the overall architecture of our model. The model mainly consists of three layers: embedding layer, graph attention layer, and predicting layer. The embedding layer initializes embeddings of nodes and edges in the graph; the graph attention layer updates the representations of nodes via GAT [17] and the predicting layer predicts sentiment polarities using the final representations of sentence and aspect nodes. The details of these three layers will be introduced in the rest of this section.

B. Embedding Layer

In the embedding layer, we initialize the embedding of nodes and edges in the graph. To be more specific, we initialize word nodes with pretrained word embeddings $X_w \in \mathbb{R}^{n*d_w}$, where d_w is the dimension of word embeddings. As for sentence nodes, we first convert a sentence s into concatenation of word embeddings. Then, we feed them into a CNN [18] layer with different kernel sizes and an LSTM layer [19] to extract local and global information of sentences, and the final representations of sentence nodes are $X_s \in \mathbb{R}^{m*d_s}$, where d_s is the dimension of sentence embeddings. The aspect nodes are initialized with one-hot vectors and then fed into a linear layer to get representations $X_a \in \mathbb{R}^{k*d_a}$, where d_a is the dimension of the aspect embeddings.

Since the same word occurring at different positions of a sentence may represent different sentiment polarities, we initialize edges E_{ws} between word and sentence nodes with position embeddings, following the position encoding approach in transformer [20].

C. Graph Attention Layer

Given the initialized aspect graph $\mathrm{G} = \{V_w, V_s, V_a, E_{ws}, E_{sa}\}$, we enhance GAT [17] with extra edge information. This

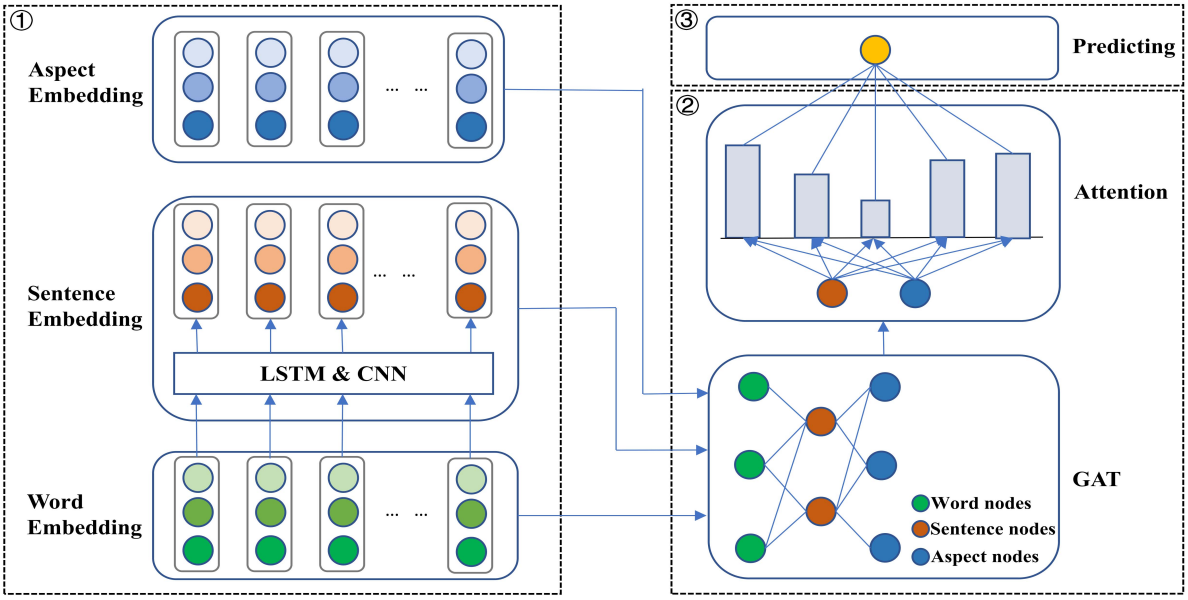

Fig. 2. Overall architecture of our HAGNN. The model consists of three layers. 1: embedding layer. 2: graph attention layer. 3: predicting layer. Also, there are three kinds of nodes: word nodes, sentence nodes, and aspect nodes, and two kinds of edges: word-sentence edges and sentence-aspect edges in the graph. The final representations of sentence nodes and aspect nodes will be used to predict sentiment polarities.

is because there are multiple aspect categories in one sentence, but not all the aspect information in this sentence is beneficial to other sentences, so we use the attention mechanism to control the information flow so that only useful information can be passed between sentences. The graph contains $m+n+k$ nodes, where m, n, and k represent the number of word nodes, sentence nodes, and aspect nodes, respectively.

As mentioned in Section I, structure patterns and semantic patterns can be learned from interactions between sentences and aspect categories. When updating, sentence nodes can learn structure-level knowledge from other similar sentence nodes through word and aspect nodes, which can strengthen features of sentences to improve the representation capacity of our model. The aspect nodes can learn knowledge of semantic diversity from sentence nodes, which can improve the generalizability of our model.

We use h_i and N_i to represent the hidden state of node i and its neighbors, respectively. The GAT layer updates the node's hidden states through a multihead attention mechanism. Also, new representation h_i^g of node i is computed by

$$h_i^g = \Big\Vert_{n=1}^{N} \sigma\left(\sum_{j\in N_i} \alpha_n^{ij} W_n h_j\right) \tag{1}$$

$$\alpha_n^{ij} = \frac{\exp\left(f\left(W_n^a\left[W_n^k h_i; W_n^v h_j\right]\right)\right)}{\sum_{l\in N_i}\exp\left(f\left(W_n^a\left[W_n^k h_i; W_n^v h_l\right]\right)\right)} \tag{2}$$

where $\Vert$ is the concatenation operation over vectors and α_n^{ij} represents the attention score between nodes i and j in the nth attention head. W_n, W_n^a, W_n^k, and W_n^v represent the learnable weights in the nth attention head. $\sigma(\cdot)$ is a nonlinear function and $f(\cdot)$ is the LeakyReLU function [21].

To add position information of edges between word nodes and sentence nodes, we modify the traditional GAT layer by adding edge information e_{ij}, which denotes the edge embeddings between nodes i and j. Also, (2) is converted into

$$\alpha_n^{ij} = \frac{\exp\left(f\left(W_n^a\left[W_n^k h_i; W_n^v h_j; e_{ij}\right]\right)\right)}{\sum_{l\in N_i}\exp\left(f\left(W_n^a\left[W_n^k h_i; W_n^v h_l; e_{il}\right]\right)\right)}. \tag{3}$$

Given the node representations at the tth iteration H_c^t and their neighbors' representations H_g^t, (1) can be simplified as

$$H_c^{t+1} = \mathrm{GAT}\left(H_c^t, H_g^t\right). \tag{4}$$

We use H_w^t, H_s^t, and H_a^t to represent representations of word nodes, sentence nodes, and aspect nodes at the tth iteration, and we initialize them as: $H_w^0 = X_w$, $H_s^0 = X_s$, and $H_a^0 = X_a$. Then, we update their representations using information from their neighbors. To improve the model performance, we also add a positionwise feedforward layer (FFN) with a residual connection after the GAT layer just as [22]. At the $(t+1)$th iteration, updating rules for three kinds of nodes can be represented as

$$U_w^{t+1} = \mathrm{GAT}\left(H_w^t, H_s^t\right) \tag{5}$$

$$H_w^{t+1} = \mathrm{FFN}\left(H_w^t + U_w^{t+1}\right) \tag{6}$$

$$U_a^{t+1} = \mathrm{GAT}\left(H_a^t, H_s^t\right) \tag{7}$$

$$H_a^{t+1} = \mathrm{FFN}\left(H_a^t + U_a^{t+1}\right) \tag{8}$$

$$U_s^{t+1} = \mathrm{GAT}\left(H_s^t, H_w^t\right) + \mathrm{GAT}\left(H_s^t, H_a^t\right) \tag{9}$$

$$H_s^{t+1} = \mathrm{FFN}\left(H_s^t + U_s^{t+1}\right). \tag{10}$$

After some iterations, sentence nodes will aggregate information from not only word nodes and aspect nodes but also other similar sentence nodes through word and aspect nodes, and thus, representations of sentence nodes and aspect nodes with structure and semantic information can be better represented and used to predict sentiment polarities.

TABLE II
STATISTICS OF THE DATASETS FOR THE ACSA TASK

Dataset	Rest2014		RestLarge		Rest2014-hard		RestLarge-hard		MAMS-small	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
Positive	2179	657	2710	1505	125	21	182	92	1000	245
Negative	839	222	1198	680	123	20	178	81	1100	263
Neutral	500	94	757	241	47	12	107	61	1613	393

D. Predicting Layer

In the predicting layer, we use the final representations of sentence nodes and aspect nodes from the graph to predict probability distributions of the sentiment polarities. Given the representation of sentence i, V_S^i, and the representations of aspect j with sentiment k, V_A^{jk}, predicting operation can be formulated as

$$V_A^j = V_A^{j1} \big|\big| V_A^{j2} \big|\big| \cdots \big|\big| V_A^{jM} \tag{11}$$

$$\hat{y}_i^j = \text{softmax}\Big(W_a V_S^i V_A^j + b_a\Big) \tag{12}$$

where $\hat{y}_i^j \in \mathbb{R}^M$ and M is the number of sentiment polarities, while W_a and b_a are learnable weights and bias, respectively. Also, the model is trained by minimizing the cross-entropy loss between the ground truth y_i^j and the output of predicting layer $\hat{y}_i^j$

$$\mathcal{L} = -\sum_i \sum_j y_i^j \hat{y}_i^j. \tag{13}$$

The algorithm of our model is shown in Algorithm 1.

Algorithm 1 Heterogeneous GAT for ACSA

Input: Initialized aspect graph $G = \{V_w, V_s, V_a, E_{ws}, E_{sa}\}$, number of iterations T, predefined aspect categories C^S mentioned in the sentence, predefined sentiment polarities P.

Output: The sentiment probability distribution $\hat{y}_i^j$ of the sentence i towards the aspect category j

// Initialization with embedding layer in Section III-B;
$H_w^0 = X_w$; // Initialize word embeddings
$H_s^0 = X_s$; // Initialize sentence embeddings
$H_a^0 = X_a$; // Initialize aspect embeddings
// Update through GAT (Equation 4) iteratively
for $t = 1, \ldots, T$ **do**
  // Equation 5 - 10
  $U_w^t = GAT(H_w^{t-1}, H_s^{t-1})$
  $H_w^t = FFN(U_w^{t-1} + U_w^t)$
  $U_a^t = GAT(H_a^{t-1}, H_s^{t-1})$
  $H_a^t = FFN(H_a^{t-1} + U_a^t)$
  $U_s^t = GAT(H_s^{t-1}, H_w^{t-1}) + GAT(H_s^{t-1}, H_a^{t-1})$
  $H_s^t = FFN(H_s^{t-1} + U_s^t)$
end for
// Predicting;
// traverse all sentences
for $V_S^i \in H_s^T$ **do**
  // traverse all aspect categories
  for aspect $j \in C^S$ **do**
    // traverse all sentiment polarities
    for sentiment $k \in P$ **do**
      // Equation 11
      $V_A^j = concat(V_A^j, V_A^{jk})$
    end for
    // Equation 12
    $\hat{y}_i^j = softmax(W_a V_S^i V_A^j + b_a)$
    Output($\hat{y}_i^j$)
  end for
end for

IV. EXPERIMENTS

A. Datasets and Experiment Settings

To evaluate the effectiveness of our model, we conduct experiments on five public datasets. Rest2014 is a restaurant review dataset from SemEval 2014 Task 4 [1]. Following [4], RestLarge merged the restaurant reviews from SemEval 2014 Task 4 [1], SemEval2015 Task 12 [2], and SemEval-2016 Task 5 [3], and incompatibilities of data are fixed during merging and data with conflict labels are removed from datasets. However, most of the sentences in Rest2014 and RestLarge contain only one aspect category, so the ACSA task degrades to a sentence-level sentiment analysis task. To measure the ability of our model to detect different sentiment polarities in one sentence toward different aspect categories, we construct the datasets Rest2014-hard and RestLarge-hard following [4], where one sentence contains at least two aspect categories with different sentiment polarities. MAMS is a dataset released in [15] and all sentences in MAMS contain at least two aspect categories with different sentiment polarities. MAMS-small is a much more difficult dataset, which is tested on the same testing set as MAMS but with less training data. Statistics of these five datasets can be found in Table II.

B. Compared Methods

To evaluate the performance of our model, we compare it with the following models, which can be divided into three groups: general sentiment analysis models, ABSA models, and BERT-based models.

1) General Sentiment Analysis Models: TextCNN [23] is a sentence-level sentiment analysis method using CNN with different kernel sizes, and it can provide a strong baseline for ACSA.

LSTM [19] is a widely used model in NLP, it performs well in sentence-level sentiment analysis, and various models are based on the LSTM architecture.

BiLSTM + AT [24] is an LSTM-based model with bidirectional architecture and attention mechanism.

2) ABSA Models: AT-LSTM [5] is an attention-based LSTM model for the ACSA task, which can capture relationships between sentences and given aspect categories.

ATAE-LSTM [5] takes the aspect information into account and appends it to the input sentence embeddings, and the attention mechanism is also used in ATAE-LSTM.

GCAE [4] is a CNN-based model for ACSA task with gating mechanisms. Since the model can be easily parallelized, it is efficient and effective.

CapsNet [15] is an aspect-level sentiment analysis method with capsule layers and performs well on many datasets.

M-AT-LSTM [25] performs the aspect-category detection task and the ACSA task simultaneously and performs well in some datasets.

As-capsule [26] uses the AS-Capsules to solve the aspect-level sentiment analysis task.

3) BERT-Based Models: BERT [27], BERT-QA-B [14], BERT-pair [27], and SCAN-BERT [8] are BERT-based models and achieve the best performance on some datasets [8].

C. Implementation Details

We provide two kinds of HAGNN called HAGNN-GloVe and HAGNN-BERT, where their difference lies in the way to initialize sentence nodes: HAGNN-GloVe uses LSTM and CNN to extract sentence representations with GloVe vectors as inputs and HAGNN-BERT use a BERT-based model [27] to extract sentence representations. The reason we provide these two variants lies in two aspects. First, some previous works, such as AT-LSTM and GCAE, did not have the BERT version, so we use HAGNN-GloVe to make a fair comparison. Second, BERT is a huge model, which may cost a lot of time and hardware resources to train and deploy, so we provide a more lightweight version HAGNN-GloVe.

For HAGNN-GloVe, we initialize word nodes with pretrained 300-D GloVe vectors [28] and initialize aspect nodes with one-hot vectors, and we initialize sentence nodes by concatenating the outputs of a two-layer bidirectional LSTM and a CNN model using GloVe vectors as inputs, the hidden size of LSTM is set to 128, the filter sizes of CNN are set to {2, 3, 4}, and the number of channels for CNN is set to 500. For HAGNN-BERT, we initialize word nodes with pretrained 300-D GloVe vectors [28] and initialize aspect nodes with one-hot vectors, and we initialize sentence nodes with a BERT-based model implemented by PyTorch and adopt its suggested hyperparameters. It should be noted that we just use BERT to initialize sentence nodes and do not update BERT while training. For both HAGNN-GloVe and HAGNN-BERT, edges between word and sentence nodes are initialized with position encoding following [29]. The learning rate is set to 0.001 and the dropout rate is set to 0.5. The number of training epochs is set to 200. We assume that the testing set is not available during training, so we use GAT in our model in an inductive-learning way. We run all methods five times and use average results on testing sets as the final results.

For compared models, inputs for non-BERT models are initialized with pretrained 300-D GloVe vectors. The filter sizes of TextCNN are set to {2, 3, 4} and the number of channels is set to 256. The hidden size of LSTM and BiLSTM + AT is set to 128 and the number of LSTM layers is set to 2. The dropout rate is set to 0.5 and the number of training epochs is set to 100. For BERT, we use a BERT-based model implemented by PyTorch and adopt its suggested hyperparameters. For rest of the models which are designed for the ACSA task, we follow the parameter settings in their original papers and refer to the implementation in [30]. All models are optimized by Adam [31].

D. Results and Analysis

Table III shows the accuracy and macroaveraged F1 score of our model and other compared models on the five public datasets. From the results, we can draw the following insights.

TextCNN, LSTM, BiLSTM + AT, and BERT perform well on the Rest2014 and RestLarge datasets but perform poorly on the other three datasets, and this is because most of the instances in Rest2014 and RestLarge datasets contain only one aspect category, so the ACSA task degrades to the sentence-level sentiment analysis task. The poor performance of these models on the other three datasets indicates that they learned little about aspect-level sentiment analysis. It also indicates that using Rest2014 and RestLarge datasets to evaluate models over the ACSA task is not close to real-world scenarios.

Other aspect-level models, AT-LSTM, ATAE-LSTM, GCAE, CapsNet, BERT-QA-B, and SCAN-BERT, also perform well on Rest2014 and RestLarge datasets and get a little improvement than the above traditional models on the Rest2014-Hard, RestLarge-Hard, and MAMS datasets. This is because they utilize the given aspect information and guide the models to attend the given aspect categories using an attention mechanism. However, they just extract information from one sentence and ignore structure and semantic interactions between sentences, which makes their improvements on the other three datasets not evident.

Our model achieves the best accuracy and macro-F1 on the RestLarge, Rest2014-Hard, RestLarge-Hard, and MAMS datasets, which indicates that our model can better handle the hard instances that have different sentiment polarities toward different aspect categories.

We contribute the reason for better performance to the following four points. First, we consider structure information between similar sentences that have the same aspect categories and sentiment polarities, and information sharing between them strengthens the performance of our model. Second, we consider diverse semantic information in different sentences, even though they have the same aspect categories and sentiment polarities. Fine-grained aspect nodes can learn this diversity and improve the generalizability of our model. We put these aspect nodes into the graph, and they can be updated with sentence nodes simultaneously, and interactions between sentences and aspect categories are more direct. Third, considering the fact that one sentence may contain many aspect categories but not all aspect information is useful to other sentences, we use a graph attention layer to control the

TABLE III

MODEL COMPARISON RESULTS (%) ON TESTING SETS. AVERAGE ACCURACY AND MACRO-F1 SCORE OVER FIVE RUNS ARE REPORTED. THE BEST SCORES ARE MARKED IN BOLD. *P*-VALUES BETWEEN HAGNN AND COMPARED METHODS ARE ALL LESS THAN 0.05, WHICH MEANS THAT THE PERFORMANCE IMPROVEMENT OF HAGNN OVER COMPARED METHODS IS SIGNIFICANT

Dataset	Rest2014		RestLarge		Rest2014-hard		RestLarge-hard		MAMS-small	
	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
TextCNN	79.49	66.29	80.07	67.77	42.64	32.38	50.15	43.04	48.79	45.64
LSTM	76.65	64.33	76.70	65.17	38.87	20.12	37.44	18.14	48.26	42.79
BiLSTM+AT	78.54	66.58	78.85	67.20	40.00	20.58	38.12	22.85	48.94	43.89
AT-LSTM	**82.39**	**71.47**	80.45	59.71	43.02	33.10	49.75	40.42	48.64	43.67
ATAE-LSTM	77.90	66.01	78.62	65.50	47.93	37.96	40.00	21.73	60.60	56.90
GCAE	77.06	64.03	79.56	66.92	40.76	22.44	48.97	34.73	64.88	63.79
CapsNet	80.70	69.91	76.04	65.78	57.74	52.34	56.92	49.82	65.26	62.41
M-AT-LSTM	81.28	70.35	80.24	65.68	53.82	50.07	56.80	48.92	63.26	61.24
As-capsule	82.17	70.30	80.37	66.20	55.02	51.27	55.82	49.15	64.21	62.50
HAGNN-GloVe	79.65	64.88	**83.14**	**69.93**	**62.26**	**53.16**	**61.28**	**57.55**	**66.92**	**65.33**
BERT	87.09	75.40	82.37	67.53	49.43	38.75	46.74	35.73	47.54	45.68
BERT-QA-B	87.43	75.52	86.62	72.73	65.43	55.24	67.82	62.55	67.23	65.88
BERT-pair	87.48	73.28	84.50	70.83	66.25	54.38	63.15	58.27	68.29	65.93
SCAN-BERT	**88.34**	76.83	86.58	72.52	65.28	55.16	64.27	57.73	71.24	67.36
HAGNN-BERT	86.20	**76.92**	**87.19**	**73.97**	**67.32**	**56.56**	**68.96**	**64.02**	**72.58**	**69.21**

information flow between sentences so that only related information can be passed to other sentences. Finally, considering the fact that the same word occurring at different positions of a sentence may represent different sentiment polarities, we use position embedding in our graph to better handle relationships between words and sentences.

V. DISCUSSION

A. Ablation Study

To better investigate the impact of different components in our model, we conduct an ablation study on the most broadly used dataset Rest2014-hard.

The results are shown in Table IV. We observe that removing word nodes or position information of edges between word and sentence nodes leads to drop on accuracy, which means that both word nodes and position information are important for the sentence nodes. The removal of sentence updating between sentence nodes also leads to worse accuracy, and we can conclude that information sharing between sentences is effective in our model. Furthermore, the removal of aspect nodes can lead to drop on accuracy, which indicates that fine-grained aspect nodes can learn semantic diversity knowledge between sentences and improve the generalizability of our model. Finally, the removal of attention mechanism in the graph leads to the largest performance drop, which shows that the attention mechanism can control the information flow between sentence nodes so that only related information can be passed to other sentence nodes.

B. Effect of GAT Layers

As a different number of GAT layers mean different receptive fields of our model, we investigate the effect of the GAT layer number L on the final performance of HAGNN. We test the value of L in set {0, 1, 2, 3, 4, 5, 6} and report the corresponding accuracy and macroaveraged F1 score on the Rest2014-hard dataset with HAGNN-GloVe. The results are shown in Fig. 3.

TABLE IV

ACCURACY (%) OF DIFFERENT MODEL VARIANTS. "-" MEANS THAT WE REMOVE THE COMPONENT FROM OUR MODEL

Model	Accuracy
HAGNN	**62.26**
- Word nodes	60.37
- Position information	58.49
- Sentence nodes	56.60
- Aspect nodes	54.72
- Attention mechanism	50.94

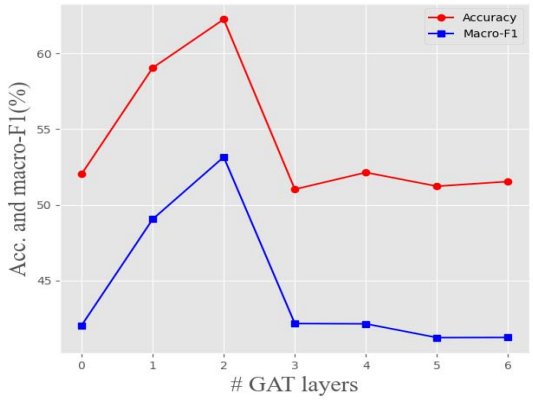

Fig. 3. Effect of the different number of GAT layers on the Rest2014-hard dataset with HAGNN-GloVe. Accuracy and macroaveraged F1 score over five runs are reported.

We can observe that when L is 0, HAGNN gets bad performance on both accuracy and macroaveraged F1 score because there is no information passing in the graph. When L is 1, aspect nodes can get semantic diversity information from sentence nodes through the graph, which is the reason

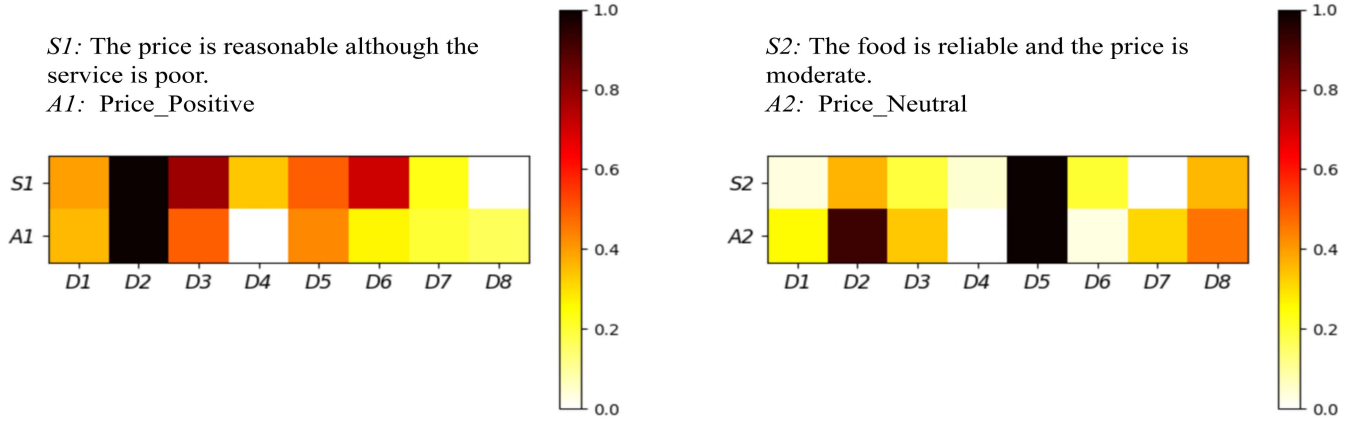

Fig. 4. Visualization of sentence embeddings and aspect embeddings on the Rest2014-hard dataset. The first line refers to sentence embeddings and the second line refers to aspect embeddings. D1–D8 refer to different dimensions of embeddings.

TABLE V
COMPARISON RESULTS (%) BETWEEN TRANSDUCTIVE LEARNING AND INDUCTIVE LEARNING WITH HAGNN-BERT

Method	Rest2014-hard		RestLarge-hard	
	Acc.	F1	Acc.	F1
Inductive	67.32	56.56	68.96	64.02
Transductive	73.24	62.49	73.89	69.12

for better performance of HAGNN. HAGNN achieves the best performance when L is 2, and this is because sentence nodes can get structure similarity information from other sentence nodes with the same aspect categories and sentiment polarities through aspect nodes by two hops. Meanwhile, aspect nodes can get diverse semantic information from sentence nodes. The above results indicate that HAGNN can learn the structure and semantic patterns between sentences through the graph and improve the performance of our model. When L increases, the performance of HAGNN drops heavily, and we think that this is because sentence nodes and aspect nodes get much noise from other unrelated sentence nodes by more than two hops.

C. Transductive and Inductive Learning

To investigate the influence of transductive learning and inductive learning on HAGNN, we conduct experiments on the Rest2014-hard dataset and RestLarge-hard dataset with HAGNN-BERT. For inductive learning, we do not add sentences in the testing set into our graph, and for transductive learning, we add sentences in the testing set into our graph as sentence nodes and link them with word nodes according to sentence-word co-occurrence. The results are shown in Table V. The results of transductive learning are much better than those of inductive learning, which is consistent with the general conclusion [32]. Also, the reason for this phenomenon is inductive learning needs models to generalize from the training set to the testing set. Even though transductive learning gets the better performance, it needs testing data while training and cannot generalize to novel samples, which is usually not in line with actual requirements, so we follow the inductive-learning way in this article.

D. Model Complexity

To analyze the model complexity, training efficiency, and inference efficiency of HAGNN, we conduct experiments on the Rest2014-hard dataset with HAGNN-BERT. From a theoretical view, the complexity of HAGNN is as same as GAT [17] since we do not update BERT during training. Thus, the model complexity of HAGNN is $O(|V|F^2+|E|F)$, where $|V|$, $|E|$, and F represent the number of nodes, the number of edges, and input feature dimensions, respectively. Taking the Rest2014-hard dataset as an example, the statics of constructed graph of our model is shown in Table VI. The number of word nodes and the number of edges can be reduced by increasing the threshold of minimum number of word appearing in sentences.

From an experimental view, we conduct experiments to analyze both training efficiency and inference efficiency of HAGNN compared with BERT. The training time for one epoch and the inference time for the entire testing set are reported in Table VII based on the average time of 50 experiments. All experiments are performed on the same hardware platform. From the table, we can see that the training efficiency of our model is higher than that of BERT since HAGNN-BERT just uses BERT to initialize sentence nodes and does not update BERT during training. The inference efficiency of our model is close to BERT since both two models need to perform feedforward propagation while inference. Furthermore, if our model performs in a transductive-learning way, the inference time of HAGNN can decrease to 0.0003 s since we have already performed feedforward propagation on the testing set during training.

E. Visualization

To better understand how HAGNN works, we illustrate the sentence embeddings and aspect embeddings to investigate how they interact with each other. We project the final representations of sentence nodes and aspect nodes in the graph to

TABLE VI

STATICS OF THE CONSTRUCTED GRAPH ON THE REST2014-HARD DATASET. $|V_w|$, $|V_s|$, $|V_a|$, $|V|$, $|E|$, AND F REFER TO THE NUMBER OF WORD NODES, NUMBER OF SENTENCE NODES, NUMBER OF ASPECT NODES, TOTAL NUMBER OF NODES, NUMBER OF EDGES, AND INPUT FEATURE DIMENSIONS, RESPECTIVELY

$\|V_w\|$	$\|V_s\|$	$\|V_a\|$	$\|V\|$	$\|E\|$	F
599	295	15	909	10636	64

TABLE VII

TRAINING TIME (SECOND) OF EACH EPOCH AND INFERENCE TIME (SECOND) ON THE ENTIRE TESTING SET FOR DIFFERENT MODELS

Model	Training / epoch	Inference time
HAGNN-BERT	15.08 sec.	5.28 sec.
BERT	29.32 sec.	5.19 sec.

the 8-D vectors through t-SNE. After normalization, we plot the weights of sentence embeddings and aspect embeddings in Fig. 4. From the figure, we can see that $S1$ and $A1$ get the maximum at the dimension 2 and get the second maximum at the dimension 3, so the attention score between them can be great enough to help our model to make right predictions. For $S2$ and $A2$, both of them achieve the maximum at the dimension 5 and achieve the second maximum at the dimension 2, so our model can easily make right predictions according to the attention score between them.

VI. CONCLUSION

In this article, we propose an HAGNN for ACSA. HAGNN can learn two types of pattern knowledge from interactions between sentences and aspect categories: structure patterns and semantic patterns, which enables structure similarity information to flow between sentence nodes with the same aspect categories and sentiment polarities and improve the performance of our model. Semantic diversity information can flow between sentence nodes and aspect nodes, which can improve the generalizability of our model. We train our model to learn representations of sentence nodes and aspect nodes simultaneously to maximize the final attention score between them. The results of experiments on five public datasets clearly show the effectiveness of our model.

REFERENCES

[1] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, and S. Manandhar, "SemEval-2014 task 4: Aspect based sentiment analysis," in *Proc. 8th Int. Workshop Semantic Eval. (SemEval)*, 2014, pp. 27–35.

[2] M. Pontiki, D. Galanis, H. Papageorgiou, S. Manandhar, and I. Androutsopoulos, "SemEval-2015 Task 12: Aspect based sentiment analysis," in *Proc. 9th Int. Workshop Semantic Eval. (SemEval)*, Denver, CO, USA, Jun. 2015, pp. 486–495.

[3] M. Pontiki *et al.*, "Semeval-2016 task 5: Aspect based sentiment analysis," in *Proc. Int. Workshop Semantic Eval.*, Jun. 2016, pp. 19–30.

[4] W. Xue and T. Li, "Aspect based sentiment analysis with gated convolutional networks," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, Jul. 2018, pp. 2514–2523.

[5] Y. Wang, M. Huang, X. Zhu, and L. Zhao, "Attention-based LSTM for aspect-level sentiment classification," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Nov. 2016, pp. 606–615.

[6] C. Zhang, Q. Li, and D. Song, "Aspect-based sentiment classification with aspect-specific graph convolutional networks," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 4567–4577.

[7] C. R. Aydin and T. Gungor, "Combination of recursive and recurrent neural networks for aspect-based sentiment analysis using inter-aspect relations," *IEEE Access*, vol. 8, pp. 77820–77832, 2020.

[8] Y. Li, C. Yin, and S.-H. Zhong, "Sentence constituent-aware aspect-category sentiment analysis with graph attention networks," in *Natural Language Processing and Chinese Computing*, vol. 12430, 2020, pp. 815–827.

[9] H. Tang, D. Ji, C. Li, and Q. Zhou, "Dependency graph enhanced dual-transformer structure for aspect-based sentiment classification," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 6578–6588.

[10] K. Wang, W. Shen, Y. Yang, X. Quan, and R. Wang, "Relational graph attention network for aspect-based sentiment analysis," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 3229–3238.

[11] S. Liu, W. Li, Y. Wu, Q. Su, and X. Sun, "Jointly modeling aspect and sentiment with dynamic heterogeneous graph neural networks," 2020, *arXiv:2004.06427*.

[12] J. Zheng, F. Cai, Y. Ling, and H. Chen, "Heterogeneous graph neural networks to predict what happen next," in *Proc. 28th Int. Conf. Comput. Linguistics*, 2020, pp. 328–338.

[13] K. Xu, H. Zhao, and T. Liu, "Aspect-specific heterogeneous graph convolutional network for aspect-based sentiment classification," *IEEE Access*, vol. 8, pp. 139346–139355, 2020.

[14] C. Sun, L. Huang, and X. Qiu, "Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence," 2019, *arXiv:1903.09588*.

[15] Q. Jiang, L. Chen, R. Xu, X. Ao, and M. Yang, "A challenge dataset and effective models for aspect-based sentiment analysis," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 6280–6285.

[16] R. He, W. S. Lee, H. T. Ng, and D. Dahlmeier, "An interactive multi-task learning network for end-to-end aspect-based sentiment analysis," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 504–515.

[17] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*.

[18] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[19] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[20] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[21] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," 2015, *arXiv:1505.00853*.

[22] D. Wang, P. Liu, Y. Zheng, X. Qiu, and X. Huang, "Heterogeneous graph neural networks for extractive document summarization," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 6209–6219.

[23] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1746–1751.

[24] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao, and B. Xu, "Attention-based bidirectional long short-term memory networks for relation classification," in *Proc. ACL*, Berlin, Germany, Aug. 2016, pp. 207–212.

[25] M. Hu *et al.*, "CAN: Constrained attention networks for multi-aspect sentiment analysis," 2018, *arXiv:1812.10735*.

[26] Y. Wang, A. Sun, M. Huang, and X. Zhu, "Aspect-level sentiment analysis using AS-capsules," in *Proc. World Wide Web Conf. (WWW)*, 2019, pp. 2033–2044.

[27] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[28] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Doha, Qatar, Oct. 2014, pp. 1532–1543.

[29] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[30] J. Zhou, J. X. Huang, Q. Chen, Q. V. Hu, T. Wang, and L. He, "Deep learning for aspect-level sentiment classification: Survey, vision, and challenges," *IEEE Access*, vol. 7, pp. 78454–78483, 2019.

[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[32] M. Bianchini, A. Belahcen, and F. Scarselli, "A comparative study of inductive and transductive learning with feedforward neural networks," in *Proc. Conf. Italian Assoc. Artif. Intell.*, Cham, Switzerland: Springer, 2016, pp. 283–293.

Wenbin An received the B.S. degree in information engineering from Northwestern Polytechnical University, Xi'an, China, in 2020. He is currently pursuing the Ph.D. degree with the School of Automation Science and Engineering, Xi'an Jiaotong University, Xi'an.

His research interests include natural language processing and transfer learning.

Feng Tian (Senior Member, IEEE) received the B.S. degree in industrial automation and the M.S. degree in computer science and technology from the Xi'an University of Architecture and Technology, Xi'an, China, in 1995 and 2000, respectively, and the Ph.D. degree in control theory and application from Xi'an Jiaotong University, Xi'an, in 2003.

He is currently with the Systems Engineering Institute, Xi'an Jiaotong University, as a Professor.

Ping Chen received the B.S. degree in information science and technology from Xi'an Jiaotong University, Xi'an, China, in 1994, the M.S. degree in computer science from the Chinese Academy of Sciences, Beijing, China, in 1997, and the Ph.D. degree in information technology from George Mason University, Fairfax, VA, USA, in 2001.

He is currently an Associate Professor of computer engineering and the Director of the Artificial Intelligence Laboratory, University of Massachusetts, Boston, MA, USA.

Qinghua Zheng (Member, IEEE) received the B.S. degree in computer software, the M.S. degree in computer organization and architecture, and the Ph.D. degree in system engineering from Xi'an Jiaotong University, Xi'an, China, in 1990, 1993, and 1997, respectively.

He did post-doctoral research at Harvard University, Cambridge, MA, USA, in 2002. He was a Visiting Professor of research with The University of Hong Kong, Hong Kong, from 2004 to 2005. He is currently a Professor with the School of Computer Science and Technology, Xi'an Jiaotong University, where he is also the Vice President.