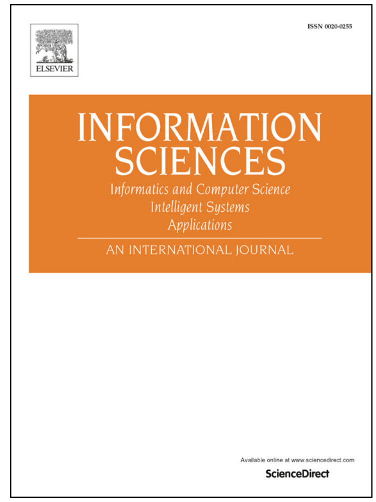
Journal Pre-proof



Dual Edge-embedding Graph Convolutional Network for Unified Aspect-based Sentiment Analysis

Chao Wu, Qingyu Xiong, Min Gao, Qiwu Zhu, Hualing Yi et al.

PII: S0020-0255(24)01437-3
DOI: https://doi.org/10.1016/j.ins.2024.121523
Reference: INS 121523

To appear in: *Information Sciences*

Received date: 10 May 2021
Revised date: 4 May 2024
Accepted date: 6 July 2024

Please cite this article as: C. Wu, Q. Xiong, M. Gao et al., Dual Edge-embedding Graph Convolutional Network for Unified Aspect-based Sentiment Analysis, *Information Sciences*, 121523, doi: https://doi.org/10.1016/j.ins.2024.121523.

Dual Edge-embedding Graph Convolutional Network for Unified Aspect-based Sentiment Analysis

Chao Wu[a,b], Qingyu Xiong[a,b,*], Min Gao[a,b], Qiwu Zhu[a,b], Hualing Yi[a,b], Jie Chen[c]

[a]*Key Laboratory of Dependable Service Computing in Cyber Physical Society (Chongqing University), Ministry of Education, China*
[b]*School of Big Data and Software Engineering, Chongqing University, Chongqing, 401331, China*
[c]*Special Equipment Inspection And Research Institute of Chongqing*

Abstract

Aspect-based sentiment analysis (ABSA) mainly involves three elements, i.e., aspect, target, and sentiment. Most existing research focuses on aspect-sentiment or target-sentiment separately, while connection among the three elements is still underexploited. In order to extract three interrelated elements from a sentence simultaneously, we propose a unified end-to-end model, which is based on one aspect via a virtual edge attention mechanism and a graph convolutional network (GCN) to detect related targets and to analyze the corresponding sentiment polarity. The virtual edges are decided by the association intensity between the aspect and the words of the sentence. Instead of naively regarding all connecting edges as binary values, we design a novel GCN with a dual edge-embedding method to help information flow among each word in the sentence and between the aspect category and the sentence. Extensive experiments on two open datasets show that our model outperforms the state-of-the-art models in the aspect-target-sentiment triples detection task and its two subtasks.

Keywords: `Aspect-based sentiment analysis, End-to-end, Graph convolutional network, Attention mechanism, Dual edge-embedding`

*Corresponding author
Email addresses: `chaowuml@gmail.com` (Chao Wu), `cquxqy@163.com` (Qingyu Xiong), `gaomin@cqu.edu.cn` (Min Gao), `qiwuzhu@163.com` (Qiwu Zhu), `yihualing@cqu.edu.cn` (Hualing Yi), `13908367672@163.com` (Jie Chen)

1. Introduction

Aspect-based sentiment analysis (ABSA) is a fundamental fine-grained task in the text sentiment analysis field, aiming to analyze the sentiment polarities of aspects or targets in a sentence. ABSA normally involves three basic elements: aspect, target, and sentiment. For example, given a comment sentence by a restaurant consumer saying "*The food was well prepared and the service impeccable.*" According to ABSA, this sentence contains two tuples ("*food*", "*food quality*", "*positive*") and ("*service*", "*service general*", "*positive*"). The "*food*" and "*service*" are targets and part of the sentence. The "*food quality*" and "*service general*" denotes aspects, which are usually pre-defined according to the domain to which the sentence belongs, and the number is generally controlled at 10 to 20. Sentiments generally are divided into three types: "*positive*", "*neutral*", and "*negative*". In this example, the sentiments of targets and aspects are both "*positive*".

Most existing studies [1–4] usually used known targets to analyze the sentence sentiment polarity, while the known targets are extracted from sentences by inefficient manual methods or topic models [5]. With the successful application of the deep learning methods on natural language processing (NLP), deep learning has been used to extract targets from sentences automatically [6–8]. Recently, constructing a graph convolution network (GCN) based on a syntactic dependency tree [9] achieved remarkable success in the sentiment analysis task [10, 11]. However, these methods judged the connectivity of nodes according to the dependent edges and ignored the dependence information between two nodes carried by different edges. To detect targets and judge sentiments in the sentence, the pipeline idea was widely used to joint two independent models [12, 13]. However, these methods neglect the strong correlation between aspects and targets, and the pipeline method stacks the errors, resulting in poor performance.

In order to consider the interdependence among the three elements simultaneously, there is a challenging task named target-aspect-sentiment detection (TASD) [14]. It aims to extract three highly correlated elements to form triples instead of only considering the relationship between two elements. Since the aspects are not part of the sentence, they cannot be extracted directly by the model. Nonetheless, we know that the number of aspects in each domain is limited. Therefore, we use a small workforce to manually generate the aspects of related domains and use them as auxiliary input information of the model.

This paper designs a novel Jointly Detection with Dual Edge-embedding GCN (JDDE-GCN) model to extract all (aspect, target, sentiment) tuples from a sentence. The model connects the embedded vectors of the word and the POS-tagging embedding vectors, while a bi-directional long short-term memory network (Bi-LSTM) is used to capture sequence information. In order to construct the information transmission channel between the aspect and sentence, a virtual attention edge method is applied based on the Bi-LSTM layer output, followed by a GCN with a dual edge-embedding mechanism which computes each node representation via relation edge information flow of syntactic and virtual. All node representation features are fed to two classifiers for sentiment analysis and target recognition.

Experiments on two open benchmarking datasets show that our model outperforms the exiting state-of-the-art model for the TASD task and its correlative two subtasks.

The main contributions of this work can be lighted as:

- We propose a novel end-to-end model to extract all aspect-target-sentiment triples from a sentence by capturing interdependence among the three elements.
- We propose to use an attention mechanism to construct virtual edges between the aspect and sentence, which is a channel for information transmission between them in subsequent GCN module.
- We design a dual edge-embedding method to encode the connection at-

tributes to optimize node representation, instead of only considering the connectivity among nodes in the GCN.

2. Related Work

ABSA is a trendy branch of text sentiment analysis, ABSA methods are roughly divided into two categories: traditional and neural network methods. Traditional methods mainly utilize machine learning algorithms based on complex feature engineering, such as conditional random field (CRF) [15] and support vector machine (SVM) [16]. As feature engineering is labor-intensive, researchers recently prefer to use end-to-end neural network methods for ABSA tasks. Wang et al. [17] used LSTM to fuse a sentence text and a target sequence, and utilized an attention mechanism to focus concentrate on different parts of the sentence according to different targets. Song et al. [18] proposed dual multi-head attention framework, which learned introspective information of sentence words, and context-perceptive information between sentence and target, respectively. Liang et al. [19] combined multi-view representation learning and heterogeneous attention mechanisms to a CNN-based model for automatically extract feature representations. Wang et al. [20] adopted a sentence-level discourse segmentation method to split a sentence into several clauses and leveraged multiple Bi-LSTM layers and a word-level attention layer to distinguish the sentiment polarity of the clauses. These neural network methods resorted to various attention mechanisms and use them to improve the performance of the model.

More recent sentiment analysis research [21–24] showed that syntactical dependency trees could help models obtain more significant performance improvement. Yao et al. [25] introduced GCN to encode syntactical dependency trees for sentiment analysis task and achieved a good result. Li et al. [26]proposed a dual graph convolutional networks (DualGCN) model, in which a SynGCN module alleviated dependency parsing errors and a SemGCN module captures semantic correlations. Wang et al. [27] reshaped and pruned standard dependency trees to an aspect-oriented tree structure and used a new graph attention network to establish the relationship between opinion words and aspects. Huang et al. [28] created a converter that allows for direct syntax interaction between any two tags. The converter used syntax to guide the token-level vector, which is then used to improve the semantic representation obtained by BERT. This enhanced the overall effectiveness of the model's representation. However, most GCN methods for sentiment analysis only denoted graph edges to binary or scalar values without considering relationship information between the head node and the tail node. On the contrary, there are many pieces of research on edge embedding representation in knowledge graphs. Based on the idea of translation and nearest neighbor, Bordes et al. [29] proposed that relationships can be thought of as translation operations from the head-to-tail entities in a low-dimensional vector space, i.e. $h + r \approx t$. This work is considered a pioneering contribution in the field of knowledge graph representation learning. The researchers Nickel et al. [30] developed a novel method that combines the expression power of RESCAL [31] with the efficiency and simplicity of DistMult [32]. They achieved this by first merging entity representations into a compressed tensor product form of composite representations, denoted as $h * t \in r^d$, through loop-related operations. The main purpose of using $*$ is to reduce complexity and store the combined representations in a compressed form. Additionally, the calculation process can be further accelerated using the Fast Fourier transform. However, in sentiment analysis, the relationship types between nodes are unstable, which cannot be learned directly using the idea of translation.

We propose a unified model which differs from the above methods. The model exploits an attention mechanism to construct the information transmission channels between aspect and sentence . Besides, the proposed model learns a dual embedding representation of edge to improve the graph convolution network.

3. Methodology

In this section, we introduce each component of the JDDE-GCN model in detail. The complete framework is shown in Fig. 1.

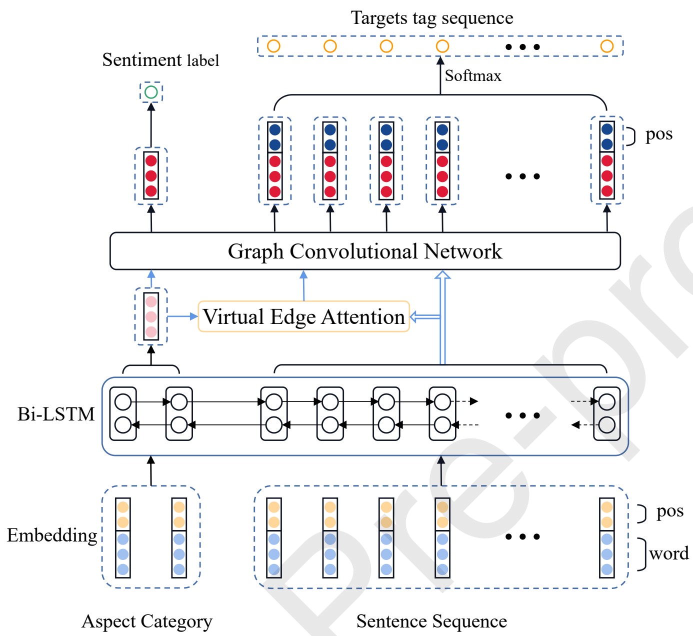


Fig. 1 The overall architecture of the JDDE-GCN model.

3.1. Problem Definition

Given a sentence $S = \{w_1, w_2, ..., w_n\}$ consisting of n tokens, a pre-identified set of aspect A and a fixed set of sentiment polarity P. An end-to-end TASD task aims to extract the three related elements target, aspect, and sentiment from S to form a tuple $(t,\ a,\ p)$, where target t is a subsequence of S. The objective of TASD task is to detect the target t associated with a specific aspect a and classify the sentiment polarity p. It is important to make sure that the sentiments corresponding to the aspect and target are all p. In this work, in order to extract the corresponding target and sentiment polarity according to a specific aspect simultaneously, we formulate a sentiment classifier and a target detector in our model. The input of our model is the combination of each aspect category and sentence, which is a sentence converted into $|A|$ sentence-aspect pairs, $|A|$ is the number of the aspect categories, as shown in the input part of Fig. 1. The target detector, a sequence labeling problem with the "TO" labels, aims to detect related targets of a specific aspect in the sentence. The sentiment classifier aims to predict a sentiment polarity $p \in \{positive, neutral, negative, N/A\}$ according to a specific aspect a and sequence S, where $p = N/A$ denotes this aspect has nothing to do with the sentence. Otherwise, if the $p \neq N/A$, we can obtain q triples (t, a, s) based on the target detector module. The results indicate that q targets are related to the aspect-sentiment pair in the sequence S. The special case when $q = 0$ represents an implicit target in the sentence to obtain one (NULL, a, s) triple.

3.2. Embedding and Bi-LSTM layers

The task input contains two parts: a sentence and a specific aspect category. Thus we convert these to the format "[CLS] + aspect category + [SEP] + sentence + [SEP]" by $[CLS]$ and $[SEP]$ tokens, where total consisting of $n+5$ tokens. Each token is transformed into a vector x_i by looking up the embedding matrix in the embedding layer. We combine the two embedding vectors along the dimension of length through the cascade method. One is obtained token embedding vector $X^t = \{x_1^t, x_2^t, ..., x_{n+5}^t\} \in \mathbb{R}^{(n+5)*d_t}$ from a pre-trained word embedding method, where d_t is the dimension of word embedding method [33]. Another POS-tagging label [1] embedding vector $X^p = \{x_1^p, x_2{}^p, ..., x_{n+5}^p\} \in \mathbb{R}^{(n+5)*d_p}$ is generated by a random initialized method, where d_t is the dimension of experiment setting. We concatenate the token embedding vector and POS-tagging embedding vector to obtain the complete embedding representation, defined as,

$$X = X^t \circ X^p = \{x_1, x_2, ..., x_{n+5}\} \tag{1}$$

where $X \in \mathbb{R}^{(n+5)*d_e}$, $\circ$ denotes vector concatenate operator, and $d_e = d_t + d_p$. A simple Bi-LSTM is used as a preliminary encode between aspect category

[1]https://universaldependencies.org/u/pos/

and sentence. We concatenate the hidden state of the backward direction and forward direction to get the related features information:

$$h_b = LSTM(\overrightarrow{X}) \circ LSTM(\overleftarrow{X}) \tag{2}$$

where $h_b \in \mathbb{R}^{(n+5)*(2d_b)}$ is the output of the Bi-LSTM layer, d_b denotes the number of unidirection LSTM hidden units. Considering that subsequent modules do not need corresponding information of $[CLS]$ and $[SEP]$, we slice h_b to get $h_i \in \mathbb{R}^{(n+2)*(2d_b)}$ according to the location information. Since one aspect category only contains two words, and some aspect categories contain the same words, such as "food quality" and "food prices," it is easy to cause ambiguity in subsequent processing. Therefore, we fuse the two words to form a whole to distinguish different aspect categories better, and we utilize the averaging method to process the aspect category to obtain the final output $B = \{b_1, b_2, ..., b_{n+1}\} \in \mathbb{R}^{(n+1)*(2d_b)}$ of the Bi-LSTM layer (here we treat each aspect category as a single unit).

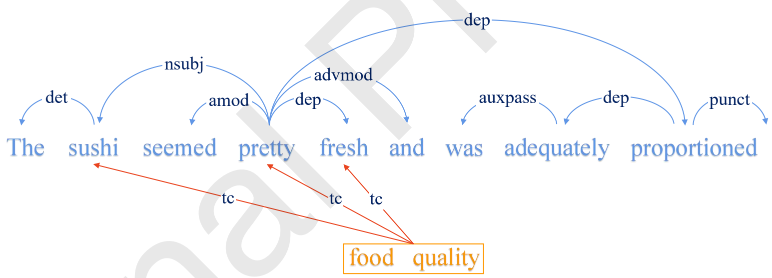


Fig. 2 Examples of syntactic dependencies with virtual edges.

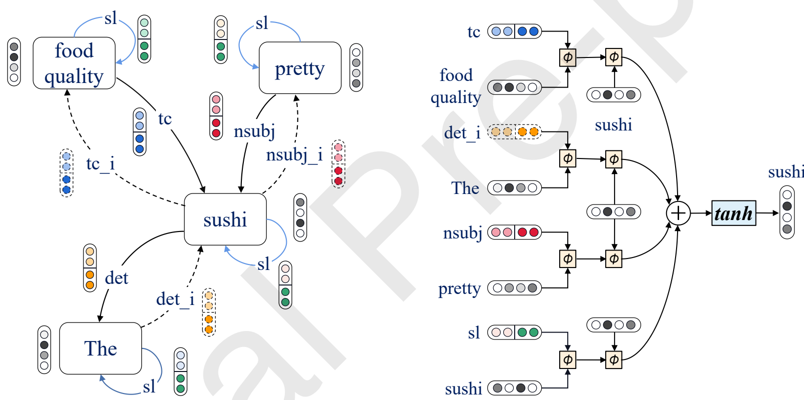
3.3. Virtual Edge Attention

In the TASD task, the aspect category is usually not part of the sentence, and each word of the sentence has a different degree of association with the aspect category. Therefore, we design a virtual edge attention mechanism to build contact between the aspect category and words with high correlation. The virtual edge value is computed as:

$$I = att(b_1, b_{2,3,...,(n+1)}) = b_1 \cdot b^T_{2,3,...,(n+1)} \tag{3}$$

$$\alpha_i = \frac{exp(I_i)}{\sum_{i=1}^{n} exp(I_i)} \tag{4}$$

where $att(\cdot)$ function represents the dot-product attention operation [34], each value of $I \in \mathbb{R}^{1*n}$ denotes the correlation between aspect category and each word of the sentence. With a softmax function, we get aspect-to-sentence attention $\alpha \in \mathbb{R}^{1*n}$. By analyzing the importance of each word of the sentence, we set a hyper-parameter threshold β. If attention value α_i greater than threshold β, we build a virtual edge "tc" between the i^{th} word of sentence and aspect category. As shown in the example in Fig. 2, during the experiment, the relevance between "sushi", "pretty" and "fresh" in the sentence, and "food quality" exceeds the threshold β, so they are connected by the "tc" edge (red lines).



(a) A partial schematic of a graph network. (b) The flow diagram of a node updating once.

Fig. 3 A local diagram of a graph convolution network.

3.4. Graph Convolution Layer

The graph convolution layer learns word representation of sentence syntactically by dependency parsing tree and assists the flow of information between sentence and aspect category. All the relationships can be represented with a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, v_2, \cdots, v_{n+1}\}$ represents each word in the B, and $\mathcal{E}$ is a set of edges. Each edge (v_i, v_j) denotes a directed connection from node v_i to v_j with an edge label $l(v_i, v_j)$. For example, in Fig. 2, each word corresponds to a node, in which "food quality" as a whole is a node in the graph. Each edge (blue and red lines) is in the set E, where "det", "tc", and so on represent the attribute label l of the edge. As there is no evidence to show that the information between two nodes only flows in one direction. Following [35], we add reversed edges and all self-loops edges to allow useful information to flow more fully and completely. As shown in Fig. 3(a), the edge ("sushi", "The") with a type label "det", the reversed edge ("The", "sushi") with a type label "det_i", and two self-loops edge of "sushi" and "The" with type label "sl".

Unlike most existing methods which simplify graph edges to binary or scalar values denoting node connectivity, our model learns a $2d_b$-dimensional embedding vector $g_{l_{ij}}$ to represent relationships from node v_i to node v_j. Each edge embedding is a fusion of two different edge embeddings $g^a_{l_{ij}} \in \mathbb{R}^{2d_b}$ and $g^b_{l_{ij}} \in \mathbb{R}^{2d_b}$. The first embedding $g^a_{l_{ij}}$ is generated by lookup a general embedding matrix, which means that the same edge types own the same embedding vector. The second embedding $g^b_{l_{ij}}$ is formed by integrating the head node, edge node, and tail node information of the upper layer.

The update method of the representation vectors of all edges in the $(k+1)$ layer is as follows:

$$g^{b(k+1)}_{l_{ij}} = f(W * (g^k_{v_i} \circ g^k_{l_{ij}} \circ g^k_{v_j}) + b) \tag{5}$$

where $W \in \mathbb{R}^{6d_b \times 2d_b}$ and $b \in \mathbb{R}^{2d_b}$ denote a weight matrix and a bias, $g^k_{v_i}$ and $g^k_{v_j}$ respectively represent the vector representation of nodes v_i and v_j in the $(k+1)$ layer, $f(\cdot)$ represents a non-linear operation function.

$$g^{(k+1)}_{l_{ij}} = mean(g^a_{l_{ij}} + g^{b(k+1)}_{l_{ij}}) \tag{6}$$

where $mean(\cdot)$ is an average pooling function. It should be noted that the first layer edge embedding is defined as equal to the general embedding, which is, $g^1_{l_{ij}} = g^a_{l_{ij}}$.

Inspired by [30] in the knowledge graph field, we utilize the circular-correlation method to update the representation vector of each node in the graph. Each node updating is defined as:

$$g_{v_j}^{(k+1)} = f\Big(\sum_{v_i \in \mathcal{N}(v_j)} \phi\big(\phi(g_{v_i}^k, g_{l_{ij}}^k), g_{v_j}^k\big)\Big) \tag{7}$$

Example of circular correlation operation:

$$c = \phi(a, b) \Rightarrow \begin{cases} c_0 = a_0 b_0 + a_1 b_1 + a_2 b_2 \\ c_1 = a_0 b_2 + a_1 b_0 + a_2 b_1 \\ c_2 = a_0 b_1 + a_1 b_2 + a_2 b_0 \end{cases} \tag{8}$$

where $\mathcal{N}(v_j)$ is a set of neighbors node of v_j for its ingoing edges, $\phi(\cdot)$: $\mathbb{R}^{2d_b} \times \mathbb{R}^{2d_b} \Rightarrow \mathbb{R}^{2d_b}$ denotes a circular-correlation operation. The update process of a single node in our model is depicted in Fig. 3(b).

3.5. Output layer

After obtaining the feature representations of the node, we divide all node representation into two parts, g_{v_1} and $g_{v_2, v_3, \cdots, v_{(n+1)}}$, in which g_{v_1} is the coded representation of the first node in the graph network.

Sentiment Classifier

In order to get sentiment polarity, we fed g_{v_1} into a fully connected layer and a softmax layer to generate the final sentiment classes distribution:

$$\hat{y^s} = softmax(W_s \cdot g_{v_1} + b_s) \tag{9}$$

We use labeled data to train sentiment classifier and utilize cross-entropy method as loss function:

$$L_s = -\sum_{i=1}^{C_s} y_i^s log(\hat{y_i^s}) \tag{10}$$

where C_s is the class number of sentiment polarity, y_i^s is the truth sentiment label in the dataset, and $\hat{y_i^s}$ represents the predicted probability that the observed sample sentiment belongs to classes i.

Target Detector

For target detect, we concatenate the feature representation $g_{v_2, v_3, \cdots, v_{(n+1)}}$ of each word and its corresponding POS-tagging embedding to obtain the final

Table 1

The statistics of datasets.

Datasets		Sentences	Aspect	Target	Positive	Neutral	Negative	NULL
Rest15	train	1118	12	1503	1083	52	368	375
	test	581	12	787	413	45	329	248
Rest16	train	1703	12	2298	1501	100	697	627
	test	586	12	750	513	42	195	209

representation of each word.

$$g^{'}_{v_2,v_3,\cdots,v_{(n+1)}} = g_{v_2,v_3,\cdots,v_{(n+1)}} \circ E_p \tag{11}$$

where $E_p \in \mathbb{R}^{n \times d_o}$ denotes POS-tagging label embedding vector. This vector is obtained by random initialization like X^p, but the dimensions of the embedded representation are different. The $\circ$ denotes vector concatenate operator, then, we fed it to the target detector as:

$$\hat{t_i} = softmax(W_t \cdot g^{'}_{v_i} + b_t) \tag{12}$$

The goal is to minimize the cross-entropy loss for all word from the truth data and the predicted data:

$$L_t = -\sum_{i=2}^{n+1} \sum_{j=0}^{C_t} t_i^j log(\hat{t_i^j}) \tag{13}$$

where t_i^j denotes the truth label of the $(i-1)^{th}$ word in the sentence, $\hat{t_i^j}$ is the predictive value of the probability that the $(i-1)^{th}$ word belongs to classes j.

Joint Learning

We joint the sentiment classifier with the target detector for training, and the loss function is as follows:

$$L_{total} = \alpha L_s + (1-\alpha) L_t + \lambda \|\theta\|_2 \tag{14}$$

where α is a balance coefficient used to adjust loss values weight of two subtasks, and λ is the regularization parameter.

4. Experiments

4.1. Datasets

We evaluate our model on the two widely-used restaurant reviews benchmark datasets from SemEval-2015 Task 12 [36] and SemEval-2016 Task 5 [37]. The statistical descriptions of these datasets are shown in Table 1, in which $NULL$ column represents implicit targets.

4.2. Implementation Details

We utilize $BERT_{base}$(uncased) as the initial word embedding method for all experiments. The uniform distribution $U(-0.15, 0.15)$ is used to initialize the POS-tagging embedding, where the dimension values in the embedding layer and the output layer are 132, and 50, respectively. The dimension value of all hidden layer vectors in the model is 300, and all weight values are initialized by the $Xavier$ normalization method [38]. We use Adam optimizer [39] with a learning rate of $2e{-}5$ and batch size 32 to train our model. The L_2 regularization decay of $1e-4$ and all dropout value are 0.5. The threshold β is $1e-2$ and coefficient α is 0.4. All the final experimental results are the mean value of the five random initialization run results. We use the F1-score as evaluation standards.

4.3. Baselines

To verify the effectiveness of our model, we compare it with the following several baseline models.

- E2E-TBSA [40] consists of two LSTMs, one of which is used to predict the combination of target and sentiment, and the other is used to define the boundary of prediction.

- DOER [41] uses a dual recurrent neural network to detect sentiment tag and target tag. A cross-shared mechanism is used to interact with information.

- BERT-pair-NLI-B [42] utilizes aspect to construct an auxiliary text sequence based on BERT, and then transforms the sentiment analysis task into a sentence-pair classification task.

- Baseline-1-f_lex [43] fuses syntactic parsing, partial semantic, POS-tagging, and lexical-semantic information constructs an NLP pipeline framework for sentiment analysis and target extract tasks.

- AddOneDim-BERT [44] utilizes the BERT architecture to construct a joint model and transform the ASD task into a multi-class classification problem.

- Hier-GCN-BERT [45] uses a lower-level GCN to construct the inner-relations among all aspect categories, and captures the relationships between sentiment polarity and aspect categories with a higher-level GCN.

- NCS [46] proposes a two-stage framework consisting of multiple Bi-LSTM and GCN models to identify a target and its sentiment polarity.

- TAS-BERT [14] combines aspect, sentiment, and sentence as the input of the model, and then uses the BERT model as the basic model to complete the TASD task.

4.4. Comparison Results

Table 2 shows the comparison results for the JDDE-GCN and all baseline models. According to different tasks, all models are mainly divided into TASD and its subtasks: target-sentiment detection (TSD) and aspect-sentiment detection (ASD).

For the TSD task, we observe that the JDDE-GCN model outperform all baselines and achieve 0.93% and 1.32% absolute performance gains than the strongest baselines on two datasets. In particular, when we do not consider the implicit target, the performance improvement of our model is more obvious.

Table 2

Comparison results of different models (%). In the TSD task, within bracket values represent the results for test sets after removing the implicit target.

Tasks	**Models**	**Rest15**	**Rest16**
TSD	DOER	56.33	65.91
	E2E-TBSA	53.00	63.10
	NCS	65.79	71.73
	TAS-BERT	66.11(64.29)	75.68(72.92)
	JDDE-GCN	**67.04(66.17)**	**77.00(76.10)**
ASD	Baseline-1-f_lex	-	63.50
	BERT-pair-NLI-B	63.67	72.70
	AddOneDim-BERT	61.67	69.79
	Hier-GCN-BERT	64.23	74.55
	TAS-BERT	**68.50**	74.12
	JDDE-GCN	68.23	**75.76**
TASD	Baseline-1-f_lex	-	38.10
	TAS-BERT	57.51	65.89
	JDDE-GCN	**58.17**	**68.08**

This phenomenon proves that the aid of aspect to detect the target and its corresponding sentiment is effective and reflects that it is more difficult to extract implicit target.

Our model also outperforms all other baseline models in the ASD task, except that it is slightly lower than the TAS-BERT model on the Rest15 dataset. The reasonable reason is that the JDDE-GCN model use a virtual edge attention mechanism to reasonably associate an aspect and the words in the sentence, facilitating the flow of related sentiment information.

As indicated, there are only two models as baselines for the TASD task due to less research work on detecting three elements simultaneously. We can see that the JDDE-GCN model achieves the best results on two datasets. The Baseline-1-f_lex uses an NLP pipeline method to obtain linguistic features and then extract three elements. The pipeline method extracts each element separately. Therefore, it cannot consider the relationship between the three elements. Besides, the pipeline method also causes the accumulation of errors, resulting in unsatisfactory results. Compare with TAS-BERT, which is also a unified model, our model is better. On the one hand, we build information transmission channels between aspect and sentence. On the other hand, our model captures the associate information among three elements using the GCN network. Moreover, the TAS-BERT model uses the combination of aspect and sentiment as the input information, which leads to the reconstructed dataset about three times the scale of ours. Due to the scale between TAS-BERT and

Table 3
Ablation study. w/o denotes without.

Model	**TSD**	**ASD**	**TASD**
-w/o POS	65.89/75.76	67.96/75.63	57.32/67.46
-w/o Att	66.68/76.23	67.93/74.99	57.86/67.62
-w/o GCN	63.12/74.67	64.27/72.37	54.58/65.27
-w/o Emb	66.07/75.84	67.43/75.01	57.38/67.37

Table 4

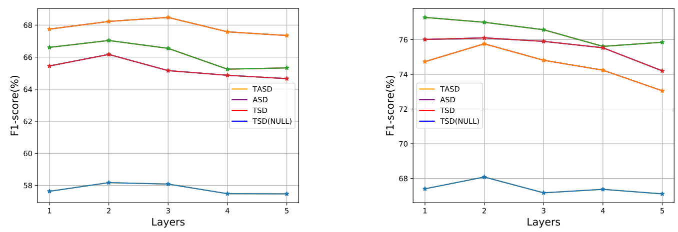
Case study. The red words represent the words connected by the virtual edges in our model.

Examples	**Baseline-1-f_lex**	**TAS-BERT**	**JDDE-GCN**
Nice ambience, but highly overrated place.	place ambience general ✗ positive ✗	place restaurant general negative	place restaurant general negative
Nice ambience, but highly overrated place.	ambience ambience general positive	ambience ambience general negative ✗	ambience ambience general positive
The lava cake dessert was incredible and I recommend it.	cake ✗ food quality positive	cake ✗ food quality positive	lava cake dessert food quality positive
A large is $20, and toppings are about $3 each.	toppings food quality negative	toppings food quality negative	toppings food quality negative

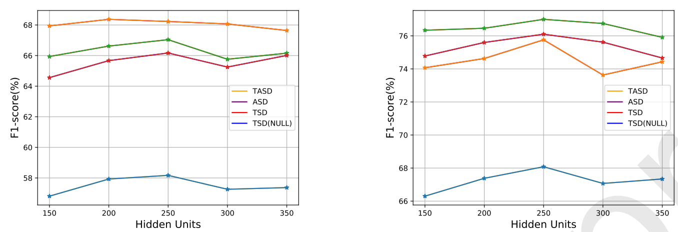
our model being very close, the TAS-BERT model takes about three times as long to train and test as our model.

4.5. Ablation Study

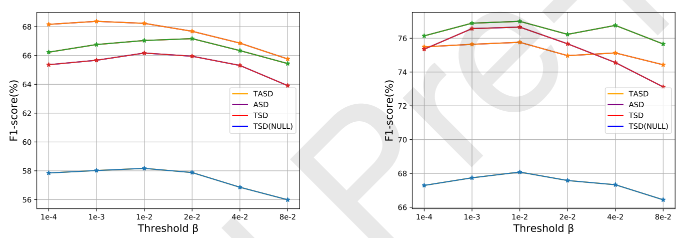
In order to further investigate the effectiveness of each crucial component in our model, we conduct several ablation experiments. The experimental results are shown in Table 3. We can see that the performance decrease of the TSD task is more notable than the ASD task after removing the POS module. This phenomenon shows that the POS tag possesses a significant impact on target detection. Removing the attention mechanism means building a virtual edge between each word of the sentence and aspect. As shown in the Table 3, the performance of the model is reduced, which indicates that the attention mechanism effectively controls the information interaction between sentence and aspect, and reduces the interference of noise information. When the GCN layer is



(a) GCN layers



(b) Hidden Units



(c) Threshold β

Fig. 4 Effect of different hyper-parameter values.

removed, the performance of the model dropped sharply on all datasets, which indicates that the syntactical dependency of the GCN layer is helpful to the feature extraction of the model. Furthermore, we remove the edge embedding mechanism in the GCN layer and only consider the connectivity between nodes. The experimental results are basically between the removed GCN layer model and the complete model, which shows that edge information can improve node feature representation performance.

4.6. Hyper-parameter Analysis

4.6.1. Number of GCN layers

Since our model involves an K-layer GCN, we study the effect of a different number of GCN layers on the model performance. Basically, we experiment with the value of K from 1 to 5 and check the corresponding F1-score on all datasets. The complete performance results are shown in Fig. 4. As indicated, when the number of GCN layers is 2, our model achieves the best performance on six of eight related tasks. Simultaneously, we also notice that when the number of GCN layers exceeds 2, the performance of the model has a noticeable downswing with the increase of GCN layers. The possible reason for this phenomenon is that with the increase of the GCN layer number, the parameters of the model increase rapidly, making the model difficult to train and easy to over-fitting.

4.6.2. Impact of hidden units number

The number of hidden units is crucial for our model because we use the same number of hidden units in the whole model. In order to investigate its effect on the performance of the model, we first vary d_b in the range of [150, 350] stepped by 50 while the number of GCN layers is 2. The complete experimental results are shown in Fig. 4. As shown, when the number of hidden units is set to 250, the model achieves the best results on most tasks. As the number of hidden units increases, the space and time cost of the model get a rapid promotion, so $d_b = 250$ is a very balanced and excellent result.

4.6.3. Impact of the threshold β

The threshold β control the number of virtual edges between aspect category and sentence. The β with a large degree indicates the model has fewer virtual edges, which means less information exchange between aspect category and sentence. As shown in Fig. 4, we present the results for different threshold β where $\beta = \{1e-4, 1e-3, 1e-2, 2e-2, 4e-2, 8e-2\}$. As threshold β increases until $\beta = 1e-2$, the results of multiple tasks all slightly rise. This is an obvious sign of the effectiveness of our virtual edge attention mechanism. When β is tiny, many virtual edges will cause some redundancy, and the model is prone to overfitting. Meanwhile, we also observe that the experimental results significantly decline as β exceeded $1e-2$. The reason for the phenomenon is that there are few or no virtual edges between aspect category and sentence when β exceeds $2e-2$. This makes the information between aspect category and sentence unable to interact effectively, resulting in a sharp decline in performance.

4.7. Case Study

Table 4 gives several examples of the TASD task. As observed in the first and second rows, the same sentence contains two different tuples. There are many errors in the Baseline-1-f_lex model. The reason is that the model does not consider the correlation among the three elements, which causes an incorrect correspondence between two tuple elements. As shown in the third line, there are errors in the target extraction of both baseline models. The main reason is that the two models ignore the syntactic relationship in the sentence, which leads to incomplete target extraction. This also verifies that syntactic relations can improve the performance of the TASD task. Seemingly, all models get the correct results in the last example. However, after analyzing the original dataset, we find that the (NULL, food quality, negative) tuple is not detected. The main reason is that the aspect and sentiment elements of the two tuples are the same, which causes the implicit target to be ignored. Since this is beyond the scope of this paper, we will concentrate on this problem in future work.

5. Conclusion and Future work

In this paper, we propose the JDDE-GCN model for the ABSA task, which can extract all (aspect, target, sentiment) tuples from a sentence. A virtual edge attention layer is implemented to build a connection between aspect and sentence to help the both information exchange. Moreover, a GCN layer with a dual edge-embedding mechanism is used to learn feature representation of the aspect and sentence. Experimental results on two open datasets show that our model outperforms the existing state-of-the-art methods for the TASD task and its subtasks. In the future, we would improve the graph attention layer by incorporating some domain-related knowledge to integrate context information better.

Acknowledgment

This work was supported by the Major Special Program of Chongqing Science & Technology Commission (No. CSTC 2019jscx-zdztzxX0031), the National Key Research and Development Program of China (Grant No. 2020YFB1712901), Graduate Scientific Research and Innovation Foundation of Chongqing (No. CYB20067).

References

[1] W. Xue, T. Li, Aspect based sentiment analysis with gated convolutional networks, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 2514–2523. doi:10.18653/v1/P18-1234.
URL https://www.aclweb.org/anthology/P18-1234

[2] F. Tang, L. Fu, B. Yao, W. Xu, Aspect based fine-grained sentiment analysis for online reviews, Information Sciences 488 (2019) 190 – 204. doi:https://doi.org/10.1016/j.ins.2019.02.064.
URL http://www.sciencedirect.com/science/article/pii/S0020025519301872

[3] I. A. Kandhro, F. Ali, M. Uddin, A. Kehar, S. Manickam, Exploring aspect-based sentiment analysis: an in-depth review of current methods and prospects for advancement, Knowledge and Information Systems (2024) 1–31.

[4] O. Alqaryouti, N. Siyam, A. Abdel Monem, K. Shaalan, Aspect-based sentiment analysis using smart government review data, Applied Computing and Informatics 20 (1/2) (2024) 142–161.

[5] S. Avasthi, R. Chauhan, D. P. Acharjya, Extracting information and inferences from a large text corpus, International Journal of Information Technology 15 (1) (2023) 435–445. doi:doi.org/10.1007/s41870-022-01123-4.

[6] X. Li, L. Bing, P. Li, W. Lam, Z. Yang, Aspect term extraction with history attention and selective transformation, in: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18, International Joint Conferences on Artificial Intelligence Organization, 2018, pp. 4194–4200. doi:10.24963/ijcai.2018/583.
URL https://doi.org/10.24963/ijcai.2018/583

[7] H. Luo, T. Li, B. Liu, B. Wang, H. Unger, Improving aspect term extraction with bidirectional dependency tree representation, IEEE/ACM Trans. Audio, Speech and Lang. Proc. 27 (7) (2019) 1201–1212. doi:10.1109/TASLP.2019.2913094.
URL https://doi.org/10.1109/TASLP.2019.2913094

[8] P. Wang, T. Liu, D. Dai, R. Xu, B. Chang, Z. Sui, Explicit interaction network for aspect sentiment triplet extraction, arXiv preprint arXiv:2106.11148.

[9] Y. Wang, Y. Ma, H. Huang, B. Wang, D. P. Acharjya, A split–merge clustering algorithm based on the k-nearest neighbor graph, Information Systems 111 (2023) 102124. doi:https://doi.org/10.1016/j.is.2022.102124.
URL https://www.sciencedirect.com/science/article/pii/S0306437922001028

[10] H. T. Phan, N. T. Nguyen, D. Hwang, Aspect-level sentiment analysis: A survey of graph convolutional network methods, Information Fusion

91 (2023) 149–172. doi:https://doi.org/10.1016/j.inffus.2022.10.004.
URL https://www.sciencedirect.com/science/article/pii/S1566253522001749

[11] T. Gu, H. Zhao, Z. He, M. Li, D. Ying, Integrating external knowledge into aspect-based sentiment analysis using graph neural network, Knowledge-Based Systems 259 (2023) 110025. doi:https://doi.org/10.1016/j.knosys.2022.110025.
URL https://www.sciencedirect.com/science/article/pii/S0950705122011182

[12] M. Hu, Y. Peng, Z. Huang, D. Li, Y. Lv, Open-domain targeted sentiment analysis via span-based extraction and classification, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 537–546. doi:10.18653/v1/P19-1051.
URL https://www.aclweb.org/anthology/P19-1051

[13] X. Li, L. Bing, P. Li, W. Lam, A unified model for opinion target extraction and target sentiment prediction, Proceedings of the AAAI Conference on Artificial Intelligence 33 (2019) 6714–6721. doi:10.1609/aaai.v33i01.33016714.

[14] H. Wan, Y. Yang, J. Du, Y. Liu, K. Qi, J. Pan, Target-aspect-sentiment joint detection for aspect-based sentiment analysis, Proceedings of the AAAI Conference on Artificial Intelligence 34 (2020) 9122–9129. doi:10.1609/aaai.v34i05.6447.

[15] B. G. Patra, S. Mandal, D. Das, S. Bandyopadhyay, JU_CSE: A conditional random field (CRF) based approach to aspect based sentiment analysis, in: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), Association for Computational Linguistics, Dublin, Ire-

land, 2014, pp. 370–374. doi:10.3115/v1/S14-2063.
URL https://www.aclweb.org/anthology/S14-2063

[16] D. Devi, C. Kumar, S. Prasad, A feature based approach for sentiment analysis by using support vector machine, 2016, pp. 3–8. doi:10.1109/IACC.2016.11.

[17] Y. Wang, M. Huang, X. Zhu, L. Zhao, Attention-based LSTM for aspect-level sentiment classification, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Austin, Texas, 2016, pp. 606–615. doi:10.18653/v1/D16-1058.
URL https://www.aclweb.org/anthology/D16-1058

[18] Y. Song, J. Wang, T. Jiang, Z. Liu, Y. Rao, Attentional encoder network for targeted sentiment classification, arXiv preprint arXiv:1902.09314doi:10.1007/978-3-030-30490-4_9.

[19] Y. Liang, H. Li, B. Guo, Z. Yu, X. Zheng, S. Samtani, D. D. Zeng, Fusion of heterogeneous attention mechanisms in multi-view convolutional neural network for text classification, Information Sciences 548 (2021) 295–312. doi:https://doi.org/10.1016/j.ins.2020.10.021.
URL https://www.sciencedirect.com/science/article/pii/S002002552031015X

[20] J. Wang, J. Li, S. Li, Y. Kang, M. Zhang, L. Si, G. Zhou, Aspect sentiment classification with both word-level and clause-level attention networks, in: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18, International Joint Conferences on Artificial Intelligence Organization, 2018, pp. 4439–4445. doi:10.24963/ijcai.2018/617.
URL https://doi.org/10.24963/ijcai.2018/617

[21] T. Sharma, K. Kaur, Aspect sentiment classification using syntactic

neighbour based attention network, Journal of King Saud University - Computer and Information Sciences 35 (2) (2023) 612–625. doi:https://doi.org/10.1016/j.jksuci.2023.01.005. URL https://www.sciencedirect.com/science/article/pii/S1319157823000058

[22] H. Wu, C. Huang, S. Deng, Improving aspect-based sentiment analysis with knowledge-aware dependency graph network, Information Fusion 92 (2023) 289–299. doi:https://doi.org/10.1016/j.inffus.2022.12.004. URL https://www.sciencedirect.com/science/article/pii/S1566253522002524

[23] W. Shang, J. Chai, J. Cao, X. Lei, H. Zhu, Y. Fan, W. Ding, Aspect-level sentiment analysis based on aspect-sentence graph convolution network, Information Fusion 104 (2024) 102143. doi:https://doi.org/10.1016/j.inffus.2023.102143. URL https://www.sciencedirect.com/science/article/pii/S1566253523004591

[24] Q. Lu, X. Sun, Z. Gao, Y. Long, J. Feng, H. Zhang, Coordinated-joint translation fusion framework with sentiment-interactive graph convolutional networks for multimodal sentiment analysis, Information Processing and Management 61 (1) (2024) 103538. doi:https://doi.org/10.1016/j.ipm.2023.103538. URL https://www.sciencedirect.com/science/article/pii/S0306457323002753

[25] L. Yao, C. Mao, Y. Luo, Graph convolutional networks for text classification, Proceedings of the AAAI Conference on Artificial Intelligence 33 (01) (2019) 7370–7377. doi:10.1609/aaai.v33i01.33017370. URL https://ojs.aaai.org/index.php/AAAI/article/view/4725

[26] R. Li, H. Chen, F. Feng, Z. Ma, X. Wang, E. Hovy, Dual graph convolutional networks for aspect-based sentiment analysis, in: Proceedings of the

59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2021, pp. 6319–6329.

[27] K. Wang, W. Shen, Y. Yang, X. Quan, R. Wang, Relational graph attention network for aspect-based sentiment analysis, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 3229–3238. doi:10.18653/v1/2020.acl-main.295.
URL https://www.aclweb.org/anthology/2020.acl-main.295

[28] X. Huang, J. Li, J. Wu, J. Chang, D. Liu, K. Zhu, Flexibly utilizing syntactic knowledge in aspect-based sentiment analysis, Information Processing and Management 61 (3) (2024) 103630. doi:https://doi.org/10.1016/j.ipm.2023.103630.
URL https://www.sciencedirect.com/science/article/pii/S0306457323003679

[29] A. Bordes, N. Usunier, A. Garcia-Durán, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, in: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, NIPS'13, Curran Associates Inc., Red Hook, NY, USA, 2013, p. 2787–2795.

[30] M. Nickel, L. Rosasco, T. Poggio, Holographic embeddings of knowledge graphs, in: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, AAAI'16, AAAI Press, 2016, p. 1955–1961.

[31] M. Nickel, V. Tresp, H.-P. Kriegel, A three-way model for collective learning on multi-relational data, in: Proceedings of the 28th International Conference on International Conference on Machine Learning, ICML'11, Omnipress, Madison, WI, USA, 2011, p. 809–816.

[32] B. Yang, W. tau Yih, X. He, J. Gao, L. Deng, Embedding entities and relations for learning and inference in knowledge bases, in: International

Conference on Learning Representations, 2014.
URL https://api.semanticscholar.org/CorpusID:2768038

[33] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. doi:10.18653/v1/N19-1423.
URL https://www.aclweb.org/anthology/N19-1423

[34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30.

[35] D. Marcheggiani, I. Titov, Encoding sentences with graph convolutional networks for semantic role labeling, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Copenhagen, Denmark, 2017, pp. 1506–1515. doi:10.18653/v1/D17-1159.
URL https://www.aclweb.org/anthology/D17-1159

[36] M. Pontiki, D. Galanis, H. Papageorgiou, S. Manandhar, I. Androutsopoulos, SemEval-2015 task 12: Aspect based sentiment analysis, in: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Association for Computational Linguistics, Denver, Colorado, 2015, pp. 486–495. doi:10.18653/v1/S15-2082.
URL https://www.aclweb.org/anthology/S15-2082

[37] M. Pontiki, D. Galanis, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, M. AL-Smadi, M. Al-Ayyoub, Y. Zhao, B. Qin, O. De Clercq, V. Hoste, M. Apidianaki, X. Tannier, N. Loukachevitch, E. Kotelnikov, N. Bel, S. M. Jiménez-Zafra, G. Eryiğit, SemEval-2016 task 5: Aspect based sentiment

analysis, in: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), Association for Computational Linguistics, San Diego, California, 2016, pp. 19–30. doi:10.18653/v1/S16-1002.
URL https://www.aclweb.org/anthology/S16-1002

[38] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks., in: Y. W. Teh, D. M. Titterington (Eds.), AISTATS, Vol. 9 of JMLR Proceedings, JMLR.org, 2010, pp. 249–256.
URL http://dblp.uni-trier.de/db/journals/jmlr/jmlrp9.html#GlorotB10

[39] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization., CoRR abs/1412.6980.
URL http://dblp.uni-trier.de/db/journals/corr/corr1412.html#KingmaB14

[40] X. Li, L. Bing, P. Li, W. Lam, A unified model for opinion target extraction and target sentiment prediction, Proceedings of the AAAI Conference on Artificial Intelligence 33 (2019) 6714–6721. doi:10.1609/aaai.v33i01.33016714.

[41] H. Luo, T. Li, B. Liu, J. Zhang, DOER: Dual cross-shared RNN for aspect term-polarity co-extraction, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 591–601. doi:10.18653/v1/P19-1056.
URL https://www.aclweb.org/anthology/P19-1056

[42] C. Sun, L. Huang, X. Qiu, Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapo-

lis, Minnesota, 2019, pp. 380–385. doi:10.18653/v1/N19-1035.
URL https://www.aclweb.org/anthology/N19-1035

[43] C. Brun, V. Nikoulina, Aspect based sentiment analysis into the wild, in: Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 116–122. doi:10.18653/v1/W18-6217.
URL https://www.aclweb.org/anthology/W18-6217

[44] M. Schmitt, S. Steinheber, K. Schreiber, B. Roth, Joint aspect and polarity classification for aspect-based sentiment analysis with end-to-end neural networks, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 1109–1114. doi:10.18653/v1/D18-1139.
URL https://aclanthology.org/D18-1139

[45] H. Cai, Y. Tu, X. Zhou, J. Yu, R. Xia, Aspect-category based sentiment analysis with hierarchical graph convolutional network, in: Proceedings of the 28th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Barcelona, Spain (Online), 2020, pp. 833–843. doi:10.18653/v1/2020.coling-main.72.
URL https://www.aclweb.org/anthology/2020.coling-main.72

[46] H. Peng, L. Xu, L. Bing, F. Huang, W. Lu, L. Si, Knowing what, how and why: A near complete solution for aspect-based sentiment analysis, Proceedings of the AAAI Conference on Artificial Intelligence 34 (2020) 8600–8607. doi:10.1609/aaai.v34i05.6383.

Declaration of Interest Statement

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☒The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Credit Author Statement

Chao Wu: Conceptualization, Data curation, Software, Writing - Original Draft
Qingyu Xiong: Writing - Review & Editing, Funding acquisition, Supervision
Min Gao: Writing - Review & Editing
Qiwu Zhu: Writing - Review & Editing
Hualing Yi: Writing - Review & Editing
Jie Chen: Writing - Review & Editing