DRGAT: Dual-relational graph attention networks for aspect-based sentiment classification

Lan You [a,d], Jiaheng Peng [a,d], Hong Jin [a,d,*], Christophe Claramunt [b,c], Haoqiu Zeng [a,d], Zhen Zhang [a,d]

[a] School of Computer Science and Information Engineering, Hubei University, Wuhan 430062, Hubei, China
[b] Naval Academy Research Institute, France
[c] Okinawa Institute of Science and Technology, Japan
[d] Engineering Research Center of Hubei Province in Intelligent Government Affairs and Application Artificial Intelligence, Wuhan 430062, Hubei, China

A B S T R A C T

Aspect-based sentiment classification has become a popular topic in natural language processing. Exploiting dependency syntactic information with graph neural networks has recently become a popular trend. Despite their success, methods that rely heavily on a dependency tree face major challenges. This concerns the alignment of aspects and their word sentiments due to the richness of the language and the fact that a dependency tree might produce noisy signals from unrelated associations. This paper introduces a Dual-Relational Graph Attention Network (DRGAT) that fully exploits syntactic structural information and then the sentiment-aware context (e.g., phrase segmentation and hierarchical structure) of the constituent tree of a sentence. Additional constituency and dependency attention mechanisms provide comprehensive syntactic information across words, thereby enabling an accurate connection between aspect words and corresponding sentiment words. Considering that the original parsed constituency tree may have a large depth, this could lead to words being far apart increasing the computational overhead. The constituency tree of each sentence is dynamically reconstructed by determining the importance of each relational node. Extensive experimental results on six English datasets demonstrated that fully exploiting syntactic information can achieve excellent sentiment classification results.

Keywords:
Aspect-based sentiment analysis
Graph attention networks (GAT)
Syntactic relations
Neural networks

1. Introduction

Aspect-based sentiment classification (ABSC) is a fine-grained task in sentiment classification that determines the sentiment polarity toward a specific aspect in a sentence [48,40]. For example, in the sentence "*The place is small and cramped but the food is fantastic,*" "*place,*" and "*food*" represent negative and positive sentiment polarity, respectively, so assigning a sentence-level sentiment polarity is inappropriate. Therefore, aspect-level sentiment analysis provides a clearer view than sentence-level sentiment analysis.

With the development of deep learning, previous studies have applied neural networks to ABSC, including Convolutional Neural Networks (CNNs) [9] and Recursive Neural Networks (RNNs) [3]. However, these approaches generally extract sentiment information from the general meaning of a sentence and word proximities and often ignore aspect-related information as precisely reflected in

* Corresponding author: *E-mail address:* anya_1024@163.com (H. Jin).

https://doi.org/10.1016/j.ins.2024.120531

a sentence. This can cause a mismatch between sentiment words and aspects, particularly when the relationship between them is semantically meaningful [4]. For example, given a restaurant review "*The food is great but the service and the environment are dreadful,*" "*service*" is closer to "*great*" rather than "*dreadful,*" and these methods may assign the unrelated sentiment word "*great*" to aspect word "*service*" mistakenly.

Other studies have leveraged syntactic structure information to build a connection between aspect and sentiment words. Early attempts were based on handcrafted rules [28], but were limited by the quantity and quality of the rules, and their generalization ability was not satisfactory. Given the massive amount of textual content, manually inferring the opinion information is a non-straightforward task [43]. Many efforts have also leveraged Graph Neural Networks (GNNs) to enhance the embedding [50,19,25]. Natural language data exhibit not only a sequential order but also an internal graph structure, such as a syntactic dependency tree or syntactic constituency tree [39]. Syntactic parsed trees (e.g., dependency trees) provide more comprehensive syntactic information [7,16]. These methods operate on the dependency tree of a sentence and the mismatching problem in long-distance sentences by building a connection between aspect words and sentiment words through a dependency tree. However, the inherent nature of the dependency tree structure may introduce noise-like irrelevant relations across clauses [7]. Furthermore, these methods typically only use syntactic information to determine whether there is an edge between nodes.

A series of recent efforts has been made to solve the mismatching problem. Chen et al. introduced an aspect-specific and language-agnostic discrete latent opinion tree model as an alternative structure to ordinary dependency trees [2]. Sun et al. enhanced the embeddings with a graph convolutional network (GCN). which is directly on the dependency tree of a sentence [12]. Zhang et al. built a GCN over the dependency tree of a sentence to exploit syntactic information and word dependencies [42]. Liang et al. were the first to exploit a constituency tree and a hierarchical structure with GNNs for ABSC. It shows superiority in the alignments between aspects and corresponding sentiment words [21]. However, they did not fully use parsed information, or specific relations between words. However, we believe that this information provides crucial clues for ABSC.

Most current ABSC studies based on Graph Neural Networks (GNNs), parsed syntactic trees are typically used for graph construction, whereas the specific syntactic relations between words are discarded. However, many types of syntactic relations between words and complex aspect expressions appear in complex sentences [32]. Because complex syntactic expressions are not properly modeled, the model's performance is suboptimal [44]. We introduce a dual-relational attention mechanism designed to improve the model's comprehension of intricate sentence structures. Inspired by [21], a dynamic approach constructs syntactically hierarchical graphs seamlessly integrating the dual-relational attention mechanism. This graph can be generated through a three-step process as outlined below: First, a method dynamically determines the importance of each relation node in each constituency tree of the sentence, called **R**elation **F**requency-**S**entence **F**requency (**RF-SF**). Non-important relation nodes are discarded before the reconstruction of the tree to obtain an appropriate depth and phrase granularity. Second, a syntax graph is reconstructed using constituency tree and dependency tree structure information. Third, given the inconsistency depth of the syntax tree of each sentence (i.e., one layer of the tree corresponds to one graph), a mapping rule is designed to combine and form hierarchical graphs. Finally, the GAT multi-head self-attention is extended, and dependency relational heads and constituency relational heads form a hierarchical **D**ual-**R**elational **G**raph **At**tention Networks (**DRGAT**) to further enhance the embeddings. The contributions of this study can be summarized as follows:

1. An extension of the original GAT model that considers two types of syntactic relational attention heads. The advantage is that the syntactic relation information between words can comprehensively model complex sentences.
2. A novel method for judging the importance of relation nodes, referred to as RF-SF. The peculiarity of this method is that it discards unimportant relation nodes and dynamically reconstructs the constituent tree of each sentence. It can effectively shorten the path distance between words and alleviate computational complexity.
3. Extensive experiments applied on six English datasets show the superiority of our model.

2. Related work

ABSC determines the sentiment polarity of a review sentence towards an opinion target. Early methods generally rely on manually defined rules [6]. However, feature engineering is labor-intensive and has almost reached a performance bottleneck.

To address the aforementioned challenges, subsequent studies replaced manually defined rules with neural network models and incorporated attention mechanisms to analyze the words surrounding the target aspect [40]. Ma et al. proposed an Interactive Attention Network (IAN) to learn attention interactively using context and target. However, common attention mechanisms are susceptible to interference from nearby sentiment words [24]. Bao et al. introduced a method for regularizing attention vectors to give the network a broader "focus" on different parts of the sentence, yet their hierarchical approach lacks consideration of multiple granularities [1]. Wang et al. highlighted the importance of considering both words and clauses in a sentence and introduced a hierarchical network that uses both word- and clause-level attention for aspect-level sentiment classification. However, syntactic information is not considered [37]. Despite the improvements in the attention mechanism employed by these methods, there is still the possibility of a mismatch when the aspects and their corresponding sentiment words are far apart.

Other studies have also combined graph neural networks (GNNs) with syntactic information. They shortened the distance between the aspects and opinion words in a sentence. Some methods employ graph convolutional networks (GCNs) with dependency trees. Graph convolutional networks (GCNs) are the most representative branch of graph neural network methods for learning representation from graph data [14]. These approaches not only reduce the distance between aspect and opinion words in a sentence but also enhance the representations [46,35]. Some methods that attempt to build different relationships between words [41,22]. However, they did not consider the constituent information.

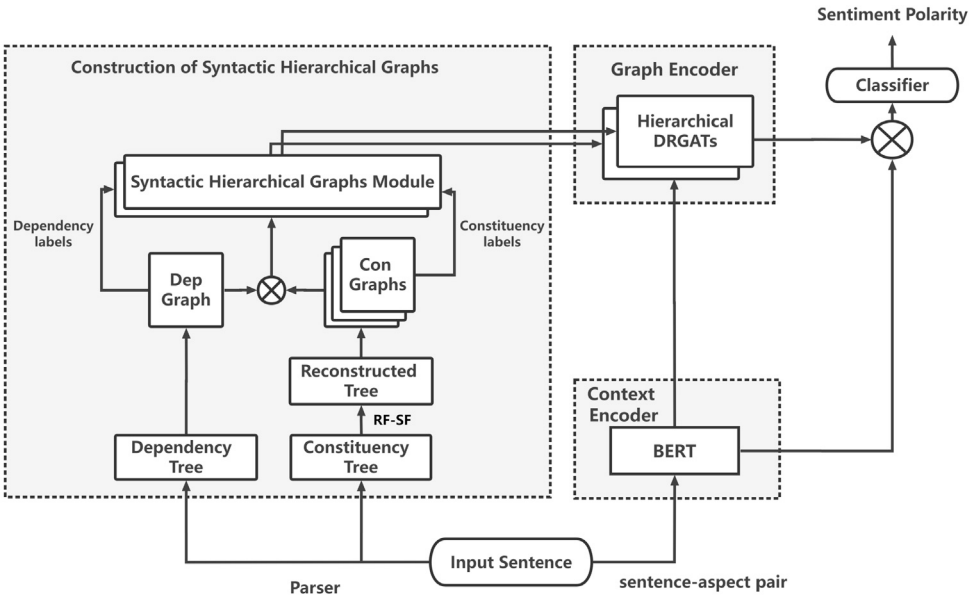


Fig. 1. The model architecture.

Syntax is a key linguistic contextual clue, as linguistic representation formalisms, the syntax can be represented by either a constituent/phrase or dependency [49]. Li et al. proposed a dual graph convolutional networks (DualGCN) model that combined syntactic and semantic features [17]. However, the specific syntactic relations between words are not considered. Liang et al. used the structural information of the constituency tree to form hierarchical graph structures, but relational nodes of the constituency tree are not taken into account (e.g., "NP," "VP," "PP," "ADJP" as illustrated in Fig. 2) into the representation space [21]. Li et al. used the constituency tree information and introduced relational nodes into the representation [18]. However, these studies did not consider syntactic relations in their attention mechanism. Wang et al. developed a relational graph attention network (R-GAT) by utilizing additional dependency relational heads, but constituency information is not considered [38]. However, we believe that simultaneously utilizing both types of syntactic relations can provide a more comprehensive perspective to the model.

Overall, most existing ABSC studies do not fully utilize syntactic (dependency and constituent) information. Specific syntactic relations between words should not be discarded because these relations provide comprehensive assistance to the attention mechanism by focusing on words that express sentiments effectively. In summary, we introduced a modeling approach that differs significantly from the methods mentioned above. First, two types of relational attention heads are supplemented to the original GAT by considering the constituency information in the attention mechanism. Second, syntactic structure information and syntactic relation information are fully exploited, and two types of syntax were considered (dependency and constituency). Third, the original parsed tree did not operate directly on the GNN. Instead, the importance of different relations is dynamically considered in different sentences, and some non-importance relations are discarded to obtain a more concise tree structure.

3. Modeling approach

3.1. Principles

3.1.1. Architecture

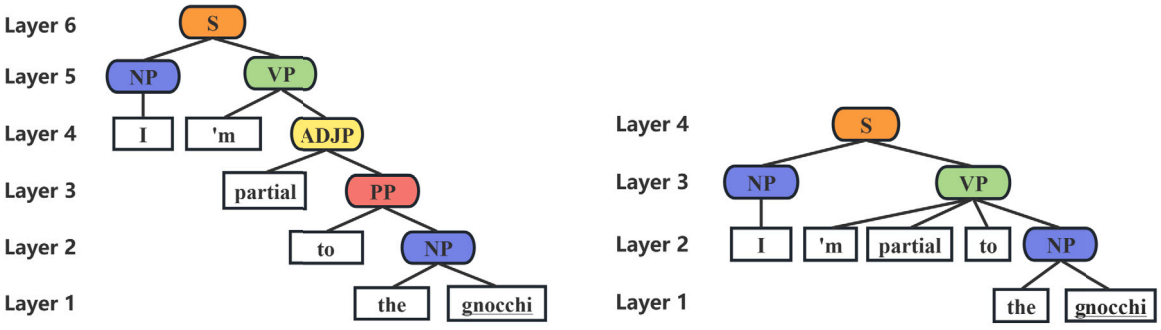
Fig. 1 illustrates the principles of the model architecture, which contains the following three key modules:

1. Context encoder that models the word embeddings with contextual semantic information.

2. Graph encoder stacked by several hierarchical DRGATs, which enhance word representations with syntactic relation information.

3. Generation of syntactic hierarchical graphs based on syntactic structure information.

Given an input sentence, the parsing tool returns the original syntactic trees. The syntactic hierarchical graphs are derived according to the following three steps: First, each constituency tree is dynamically reconstructed to discard non-importance constituents based on $RF-SF$. Second, a syntax graph was derived using a reconstructed constituency tree and a dependency tree. Third, the Syntactic Hierarchical Graphs Module selects and combines some representative graphs to form hierarchical graphs. Finally, the reconstructed constituency tree and dependency tree provide constituency and dependency information respectively, which are used to label the edges between nodes for an additional relational attention mechanism. They are then sent to the graph encoder. In the context encoder, given a sentence-aspect pair, BERT is used to obtain the textual node representations. Next, the graph encoder (i.e., hierarchical DGATs) enhances the node representations. Finally, the node representations from the context and the graph encoders are fused and sent to the classifier for sentiment classification.



(a) The original constituency tree

(b) The reconstructed constituency tree

Fig. 2. Illustration of the constituency tree.

3.1.2. Task definition

Let (S, A) denotes a sentence-aspect pair. Let $S = \{W_1, W_2, W_3, W_4, ..., W_n\}$ and $A = \{a_1, a_2, ..., a_m\}$ be a sentence and an aspect term in the sentence, respectively, where n denotes the number of words in sentence S and m the number of words of an aspect term in sentence S, with $1 \leq m < n$ and $A \in AS$ the set of aspect terms. Notably, a sentence may contain one-to-many aspect terms and each aspect term may consist of single or multiple words. The goal of ABSC is to predict the sentiment polarity $y \in \{positive, negative, neural\}$ for each $a_i \in A$.

3.2. Context encoder

We use the well-known BERT [5] to obtain word embeddings using contextual semantic information. We construct a sentence-aspect pair "[CLS] S [SEP] A [SEP]" as input to get the contextual word representations, where "[CLS]" and "[SEP]" are the special tokens in BERT. For sentences with multiple aspect terms, we input only one aspect term at a given time and use BERT multiple times. Let x_{CLS} denote the "BERT pooling" vector representation of the BERT sequence, and x_i denote the contextual representation of each token. The BERT sequence can be denoted as:

$$x = \{x_{CLS}, x_1, ..., x_n, x_{SEP}, a_i, x_{SEP}\} \tag{1}$$

For example, in the sentence "*I 'm partial to the gnocchi*", "*gnocchi*" is the aspect term. The BERT sequence is "*[CLS] I 'm partial to the gnocchi [SEP] gnocchi [SEP]*."

3.3. Syntactic hierarchical graphs construction

The above representation considers only contextual semantic information. The graph encoder integrates additional syntax information (i.e., constituency and dependency information). The graph encoder is stacked by several dual-relational graph attention networks (DRGATs).

3.3.1. Reconstruction of the constituency tree based on RF-SF

A constituency tree is a collection of labels that span over a sentence [30]. The layer with the aspect term is denoted as layer 1, and the entire constituency tree is derived from bottom to top. It is necessary to determine the appropriate granularity and balance between the number of relations and the depth of the tree to maximize efficiency. This leads us to introduce a **R**elation **F**requency-**S**entence **F**requency **(RF-SF)** to dynamically evaluate the importance of each relation in each sentence. RF represents the importance of the relation in all sentences of the entire dataset, whereas SF represents the importance of the relation in a sentence. RF-SF is computed as follows:

$$RF_{ij} = \frac{n_i * k_{ij}}{\sum_{j=1}^{J} n_i * k_{ij}} \tag{2}$$

$$SF_i = \frac{SN_i}{S} \tag{3}$$

$$(RF - SF)_{ij} = RF_{ij} * SF_i \tag{4}$$

Where SF_i denotes the proportion of sentences containing the i^{th} relation in all sentences and RF_{ij} is the frequency of the i^{th} relation in the j^{th} sentence. n_i denotes the i^{th} relation. And k_{ij} denotes the number of times the i^{th} relation appears in the j^{th} sentence. SN_i denotes the number of sentences containing the i^{th} relation and S denotes the number of sentences in the dataset.

Non-important constituents are discarded to obtain a suitable granularity. We set a simple mapping rule to obtain a reconstructed tree with a suitable granularity. Specifically, the hyperparameter α is set, which represents the threshold for judging whether the

Table 1
Abbreviation and full name of the constituents that appear in Fig. 2.

abbreviation	full name
S	Simple declarative clause
NP	Noun phrase
VP	Verb phrase
ADJP	Adjective phrase
PP	Prepositional phrase

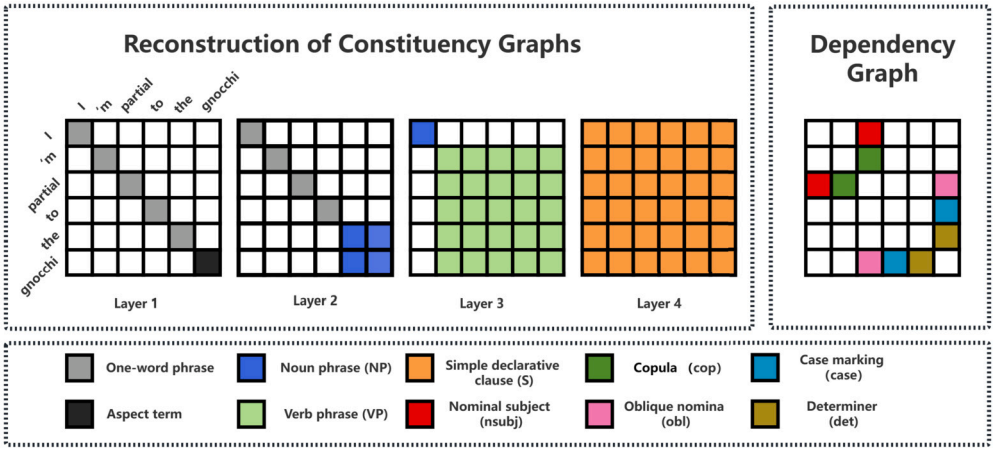


Fig. 3. Illustration of the reconstruction of constituency and dependency graphs and different colors represent different relations.

$RF-SF$ value of a relation node exhibits an important relationship. The node with an $RF-SF$ value less than α is regarded as a non-primary constituency relational node. Therefore, all the non-primary constituency relational nodes are discarded. Finally, the child nodes of these discarded nodes are connected to the retained lowest ancestor node.

Fig. 2 shows the reconstructed tree that preserves the important relational constituents, whereas Table 1 lists the full names corresponding to the abbreviations of these constituents that appear. Compared with the original tree, the number of layers (i.e., the depth of the tree) and the types of relations are reduced.

3.3.2. Syntactic graph with constituency and dependency

After reconstruction, Let $Sp=\{w_i\}_m$ denote the i^{th} word in a given phrase range and m be the number of words. For example, in layer 2 of Fig. 2b, "*the*" and "*gnocchi*" are in the same phrase range "*NP*." An input sentence consists of several phrases in each layer of the constituency tree. In each layer, $S^l=\{SP_1,SP_2,SP_3,...,SP_s\}$ indicates that a sentence consists of s phrases at the l-layer of the constituency tree. For example, $S^3=\{I,'m\ partial\ to\ the\ gnocchi\}$. For each layer of the reconstructed tree, we constructed the adjacency matrix CA, which is formulated as:

$$CA_{ij}^{l}=\begin{cases}1, & if\ w_i,w_j\ in\ the\ same\ phrase\ SP\ of\ layer\ l\\ 0, & otherwise\end{cases} \tag{5}$$

In addition to the above constituency structural information, the dependency structure information is considered. The dependency tree is treated as an undirected graph, and the adjacency matrix DA is formulated as:

$$DA_{ij}=\begin{cases}1, & if\ w_i,w_j\ are\ directly\ connected\ in\ the\ dependency\ tree\\ 0, & otherwise\end{cases} \tag{6}$$

For the reconstruction of the constituency tree, each layer of the tree corresponds to a graph, as shown in Fig. 3. Following previous work [21], several options to fully exploit syntactic structural information have been explored. One of them is finally chosen as our construction approach for the syntactic graph. Let FA denote a new adjacency matrix for each layer that corresponds to the graph. The resulting operations are given as follows:

- **Structure A. Only reconstructed constituency** $FA=CA$ This operation only considers the hierarchical reconstructed constituency graphs.
- **Structure B. Only dependency** $FA=DA$ This operation only considers the original dependency graph. We stack several dependency graphs. The number of stacked dependency graphs corresponds to the number of layers of the reconstructed constituent tree.

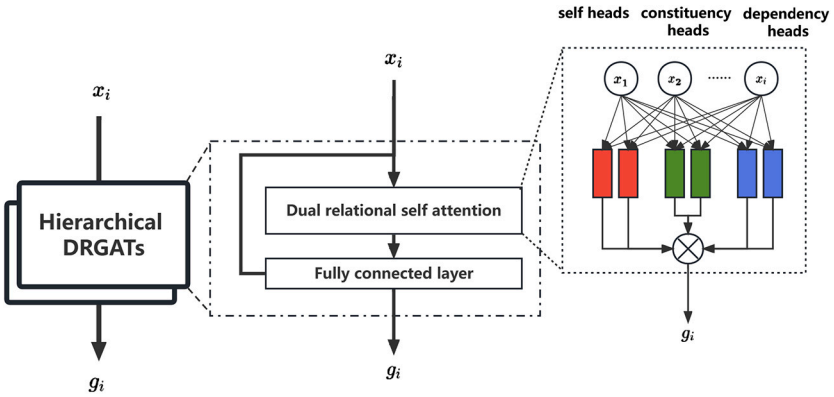


Fig. 4. Graph encoder architecture.

- **Structure C. Position-wise dot.** $FA = CA \cdot DA$ At each layer of the reconstructed tree, this operation only considers two words that are both in the same phrase and have a direct link in the dependency tree.
- **Structure D. Position-wise add.** $FA = CA + DA$ For each layer of the reconstructed tree, this operation considers two words that are in the same phrase or that have a direct link in the dependency tree.
- **Structure E. Conditional position-wise add.** $FA = CA \bigoplus DA$. For each layer of the reconstructed tree, it first deletes all dependency edges between phrases and then conducts a **position-wise add operation** with the remaining dependency edges. This operation primarily considers the removal of irrelevant noise edges between phrases.

3.3.3. Syntactic hierarchical graphs module

As shown in Fig. 3, the different layers in the constituency tree represent the different constituents and phrase granularities of a sentence. We stacked several DRGATs to extract hierarchical syntactic graphs from fine- to coarse-grained graphs. Providing a comprehensive view of GNNs should improve node representation. As the depth of the constituency tree of each sentence varies, the number of hierarchical graphs also varies. However, the number of DRGATs is fixed for each sentence. This leads us to set up a mapping rule to select some representative graphs and combine them to form hierarchical graphs. This is formulated as follows:

$$HA = \begin{cases} \{FA_j \mid j = 1 \ or \ j = 2i, i \in \{1, 2, 3, ..., LS - 1\}\} & if \ LS < n \\ \{FA_j \mid j \in \{1, 2, 3, ..., LS\}\} & if \ LS = n \\ \{FA_1, FA_2, FA_3, ..., FA_n\} + \{FA_j \mid j \in \{n+1, n+2, n+3, ..., LS\}, FA_j = FA_1\} & if \ LS > n \end{cases} \tag{7}$$

Let HA denote the collection of selected hierarchical graphs. We set the number of DRGATs as LS. Let $\{FA_1, FA_2, FA_3, ..., FA_n\}$ denote a collection of FA, where n denotes the number of graphs. FA_i represents the i^{th} layer which is selected. The cases are as follows:

- The number of candidate graphs ($\{FA_1, FA_2, FA_3, ..., FA_n\}$) is greater than the number of DRGATs ($LS < n$). First, the graph corresponding to the first layer of the tree is selected, that is, the layer where the aspect term is located. Then, from bottom to top, the corresponding graphs at intervals were selected. For example, we select layer 2, layer 4, layer 6, etc. Finally, the selected graphs are combined to form hierarchical graphs.
- The number of candidate graphs was smaller than the number of DRGATs ($LS > n$). We repeatedly stack the graph corresponding to layer 1. The layer in which the aspect term is located is assumed to be most representative. Finally, they are combined with the original candidate graphs to form hierarchical graphs.
- When the number of candidate graphs equals the number of DRGATs ($LS = n$), we use the candidate graphs directly as the hierarchical graphs.

3.4. Dual-relational graph attention networks

GAT aggregates the representations of neighborhood nodes along the graph paths [36]. Previous studies have exploited only syntactic structure information; this process does not take specific relations into account. Specific relations exist between words in the syntactic parsed trees and these specific relations cannot be ignored. The aforementioned hierarchical graphs are considered from the phrase level. However, inside a phrase, there are many syntactic relations between words (such as dependencies and constituencies). We propose extending the self-attention heads in GAT by adding two types of syntactic relational heads: constituency and dependency heads. The framework of the graph encoder of the model (i.e., hierarchical DRGATs) is shown in Fig. 4.

For each layer of the reconstructed tree, the constituency relational labeled edge describes the constituency to which two words belong. Specifically, if an edge exists between two words in the reconstructed constituency graph, the edge with a specific relation is labeled. For example, in layer 2 of Fig. 2a the label of the edge between "*the*" and "*gnocchi*" is "*NP.*" We used constituency relational heads as relation-wise gates to control the information flow from the neighborhood nodes. First, constituency relations are mapped

into vector representations. Specifically, we obtained a 200-dimensional vector representation for each specific relational label via the lookup table. Subsequently, a linear transformation layer was incorporated both preceding and subsequent to the embedding process. The constituency heads are then computed as:

$$h_{con_i}^{l+1} = \|_{k=1}^{K} \sum_{j \in N} \alpha_{ij}^{lk} W_k^l x_j^l \tag{8}$$

$$\alpha_{ij}^{lk} = \frac{exp(\sigma(relu(c_{ij} W_{k1} + b_{k1}) W_{k2} + b_{k1}))}{\sum_{j=1}^{N_i} exp(\sigma(relu(c_{ij} W_{k1} + b_{k1}) W_{k2} + b_{k1}))} \tag{9}$$

Where $h_{con_i}^{l+1}$ represents the constituency relational attention head of the node i in layer $l+1$, $\|$ denotes the vector concatenation operation. x_j^l is the vector of node i in layer j, W_k^l denotes its corresponding transformation weight matrix. α_{ij}^{lk} is the k^{th} relational attention score from node i to node j in layer l. c_{ij} denotes the constituency vector of the labeled edge between node i and node j.

The dependency relational head is similar to the constituency relational head. The formula can be expressed as:

$$h_{dep_i}^{l+1} = \|_{m=1}^{M} \sum_{j \in N} \beta_{ij}^{lm} W_m^l x_j^l \tag{10}$$

$$\beta_{ij}^{lm} = \frac{exp(\sigma(relu(d_{ij} W_{m1} + b_{m1}) W_{m2} + b_{m1}))}{\sum_{j=1}^{N_i} exp(\sigma(relu(d_{ij} W_{m1} + b_{m1}) W_{m2} + b_{m1}))} \tag{11}$$

Where d_{ij} is the mapping vector of dependency labeled edge from node i to node j, β_{ij}^{lm} denotes the m^{th} constituency relational attention score from node i to node j in layer l.

Similarly, our model also includes P original GAT self-attention heads. GAT iteratively updates each node representation (e.g., word embeddings) by aggregating neighborhood node representations using multi-head attention:

$$h_{self_i}^{l+1} = \|_{p=1}^{P} \sum_{j \in N} \gamma_{ij}^{lp} W_p^l x_j^l \tag{12}$$

$$\gamma_{ij}^{lp} = \frac{exp(\sigma(a^p \left[W^p h_i \| W^p h_j\right]))}{\sum_{k \in N_i} exp(\sigma(a^p \left[W^p h_i \| W^p h_k\right]))} \tag{13}$$

a^p is the attention mechanism vector in layer p, and W^p is the weight matrix in layer p for feature transformation. h_i and h_j are the feature vectors of node i and node j, respectively. N_i is the neighborhood of node i.

DRGAT contains K constituency heads and M dependency heads and also contains P self-attention heads of the original GAT. Let $h_{self_i}^{l+1}$ denote the self-attention head of the node i in layer $l+1$.

$$x_i^{l+1} = h_{self_i}^{l+1} \parallel h_{dep_i}^{l+1} \parallel h_{con_i}^{l+1} \tag{14}$$

$$g_i^{l+1} = relu(x_i W_{l+1} + b_{l+1}) \tag{15}$$

Where g_i^{l+1} denotes the representation of node i obtained from one DRGAT. Our graph encoder is stacked with several DRGATs. Several stacked DRGATs use the output of the previous DRGAT as the input. This can be formulated as follows:

$$g_i^{l+2} = FC(g_i^{l+1} + g_i^{l+2}) \tag{16}$$

Where FC is a fully connected feed-forward network. g_i^{l+2} is the representation of node i obtained from the 2^{nd} stacked DRGATs. Let g_i denote node i of the graph encoder. The final representation O_i can be computed as follows:

$$O_i = [x_i + g_i; x_{CLS}] \tag{17}$$

3.5. Model training

The outputs of the context and graph encoders are combined to obtain the final representations. They are sent to a fully connected layer with a softmax activation function to form the probabilities of the three sentiment polarities. They can be formulated as follows:

$$p(i) = softmax(W_i O_i + b_i) \tag{18}$$

Where W_i and b_i are trainable parameters of the classifier. The loss is the standard cross-entropy for our objective function:

$$L(\theta) = -\sum_{(S,A) \in D} \sum_{A \in AS} loss(p(i), y(i)) \tag{19}$$

Let AS denote the predefined aspect set, where A denotes the aspect term in the corresponding sentence S, and θ represents model parameters.

Table 2
The distribution of sentiment polarity and the number of constituency types in six datasets. "Pos", "Neu" and "Neg" represent the counts of positive, neutral, and negative sentiment polarities, respectively. "Con" denotes the number of constituent relation types in the constituency tree.

Dataset	Category	Pos	Neu	Neg	Con
Lap14	Train	937	455	851	25
	Test	337	167	128	25
Res14	Train	2164	637	807	24
	Test	727	196	196	24
Res15	Train	912	36	256	25
	Test	326	34	182	25
Res16	Train	1240	69	439	25
	Test	469	30	117	25
Twitter	Train	1561	3127	1560	27
	Test	173	346	173	27
MAMS	Train	3380	5042	2764	26
	Valid	403	604	325	26
	Test	400	607	329	26

4. Experiments

4.1. Datasets and setup

We evaluated our models on six English datasets: Lap14 (Laptops 14), Res14 (Restaurants 14) datasets from SemEval2014 (Task 4) [15], Res15 (Restaurants 15) from SemEval2015 [26], Res16 (Restaurants 16) from SemEval2016 [27], MAMS [13], and Twitter [7]. The Lap14, Res14, Res15 and Res16 datasets contained both multi-aspect and single-aspect sentences. Each sentence in the MAMS contained at least two aspects with different sentiments. Twitter contains sentences with only one aspect.

SuPar was used as a parser. Specifically, we used the CRF constituency parser [45] to obtain the constituent tree while using the deep Biaffine Parser [8] to obtain the dependency tree. Our context encoder is a BERT-base-uncased model. This experiment was based on the PyTorch deep learning framework version 1.7.1, with Python version 3.7. The server used in this experiment ran on a CentOS 7 operating system with 48 GB of memory. The graphics card used was a Tesla V100-PCIE with 32 GB of VRAM. The hidden layer vector dimension in BERT is set to 768. Let us set the hyperparameter $\alpha \in (0.4,0.7)$. In the Lap 14, the value of α is set to 0.45, while in the Res 14, it is set to 0.5. For the Res 15, MAMS, and Tweets, α is set at 0.65, and in the Res 16, it is set to 0.6. Adam optimizer was adopted with a learning rate of 10^{-5} and L_2 regulation of 10^{-5} for model training. The number of hierarchical DRGAT is in the range [2,3], and each DRGAT consists of two internal layers. "Accuracy" and "Macro-Averaged F1" are evaluation metrics. We applied early stopping for model training. The distribution of sentiment polarity and the number of constituency types in different datasets are listed in Table 2.

4.2. Baselines

We considered three baseline categories: 1) Neural networks-based methods; 2) Graph neural network-based methods; and 3) Graph neural network-based methods; and used BERT to obtain the embeddings.

1. Networks-based methods:

TD-LSTM [31] models the relevance of a target word to its context words and selects relevant parts of the context to infer the sentiment polarity towards the target.

ATT-BiLSTM [23] induces the attention value of the entire sentence. The model was further extended to distinguish left and right contexts, given a specific target.

AOA [10] models aspect and sentence jointly, and explicitly captures the interaction between aspect and context sentence.

IAN [24] learns attention interactively in context and target and generates representations of target and context, respectively, which can well represent the target and its collocation context.

RAM [3] adopts a multiple-attention mechanism to capture sentiment features separated by a long distance, making it more robust against irrelevant information.

LSTM uses basic LSTM for sentiment classification.

MemNet [33] introduces a deep memory network for aspect-level sentiment classification that which explicitly captures the importance of each context word when inferring the sentiment polarity of an aspect.

BERT [29] uses pure fine-tuned Bert for sentiment classification.

2. Graph-based methods:

ASGCN-DT [42] builds directional dependency graphs to exploit syntactic structure information and word dependencies.

ASGCN-DG [42] builds un-directional dependency graphs to exploit syntactic structure information and word dependencies.

GAT leverages stacked masked self-attention layers to assign different weights to different nodes in nodes' neighborhoods.

Table 3
Main experimental results on six datasets. The best results of each dataset are in **bold**, while the second-best results of each dataset are *italics*.

Category	Models	Lap14		Res14		Res15		Res16		MAMS		Tweets	
		ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1
Networks	TD-LSTM	78.00	66.73	71.83	68.43	76.39	58.70	82.16	54.21	68.02	64.83	-	-
	ATT-BiLSTM	70.39	64.83	78.21	68.34	77.15	57.66	86.35	64.01	73.25	71.69	-	-
	AOA	72.62	67.52	79.97	70.42	78.17	57.02	87.50	66.21	-	-	72.30	70.20
	IAN	72.05	67.38	79.26	70.09	78.54	52.65	84.74	55.21	-	-	72.50	70.81
	RAM	72.08	67.43	78.48	68.54	79.98	60.57	83.88	62.14	-	-	70.09	66.48
	LSTM	69.30	63.10	78.10	67.50	77.40	55.20	86.80	63.90	-	-	69.60	67.70
	MemNet	70.60	65.20	79.60	69.60	77.30	58.30	85.40	66.00	-	-	71.50	69.00
	BERT	77.59	73.28	84.11	76.68	83.48	66.18	90.10	74.16	77.56	76.13	75.52	73.23
Graph	ASGCN-DT	74.14	69.24	80.86	72.19	79.34	60.78	88.69	66.64	77.56	76.13	75.52	73.23
	ASGCN-DG	75.55	70.50	80.77	72.02	79.89	61.89	88.99	67.48	76.50	75.10	72.15	70.40
	GAT	73.04	68.11	78.21	67.17	-	-	-	-	-	-	71.67	70.13
Graph&BERT	DualGCN	*81.80*	*78.10*	**87.13**	**81.16**	-	-	-	-	-	-	77.40	76.02
	InterGCN	77.86	74.32	82.23	74.02	-	-	-	-	-	-	-	-
	SDGCN	81.35	74.32	82.23	74.01	-	-	-	-	-	-	-	-
	SAGAT	80.37	76.94	85.08	77.94	-	-	-	-	-	-	75.40	74.17
	AGCN	79.94	76.52	82.77	73.29	82.84	65.08	88.80	67.65	-	-	75.43	74.11
	DGEDT	79.80	75.60	86.30	80.00	84.00	71.00	*91.90*	*79.00*	-	-	77.80	75.40
	dotGCN	81.03	*78.10*	86.16	*80.49*	*85.24*	*72.74*	**93.18**	**82.32**	84.59	84.44	*78.11*	*77.00*
	RGAT	78.05	74.14	85.56	78.95	80.83	64.17	88.92	70.89	78.97	78.01	74.71	74.21
	BiSyn-GAT	80.37	77.06	85.07	79.26	82.81	69.53	89.51	69.98	*84.85*	*84.53*	73.85	72.95
Ours	DRGAT	**81.96**	**78.57**	86.35	79.92	**85.47**	**73.24**	91.75	78.77	**85.35**	**84.71**	**78.84**	**77.73**

3. Graph and BERT-based methods:

DualGCN [17] simultaneously considers the complementarity of syntactic structure and semantic relevance and uses two GCN modules to learn this knowledge.

InterGCN [20] builds heterogeneous graphs for each instance by exploiting aspect-focused and inter-aspect contextual dependencies for a specific aspect.

SDGCN [47] proposes a GCN-based model that can capture sentiment dependencies among multiple aspects in a sentence.

SAGAT [11] exploits syntactic awareness to model by GAT on the dependency tree structure and external pre-training knowledge by BERT.

dotGCN [2] builds an aspect-specific discrete latent opinion tree model that can materialize a connection between attention scores and syntactic distances, inducing trees from attention scores.

AGCN [46] adopts an interactive attention mechanism between aspect embeddings learned from GCN and opinion semantic embeddings learned from three Bi-LSTM.

DGEDT [34] proposes a dependency graph-enhanced dual-transformer network by jointly considering the flat representations learned from the Transformer and graph-based representations from the dependency graph in an iterative interaction manner.

RGAT [38] leverages marked dependency edges to extend the original GAT, adding relational heads to the original multi-self-attention heads.

BiSyn-GAT [21] leverages the syntactic knowledge of the constituency tree to learn the features of the nodes using the hierarchically stacked GAT layers.

4.3. Main results

4.3.1. With baselines

The experimental results for all datasets are listed in Table 3. The findings are as follows:

1) The model based on the graph neural network is generally better than neural network-based models, which shows that syntactic structure information is helpful.

2) Among the neural network-based models, BERT showed the best results. Regarding GNN-based models, the BERT-based model performed better than the others, demonstrating the effectiveness of BERT.

3) When compared to other models, dotGCN and RGAT perform better, proving that the aspect-specific pruning method is helpful.

4) BiSyn-GAT compared with other GAT-based models, proving hierarchical structure and spans can reduce noise information.

5) Regarding DotGCN comparison with other GCN-based models, and RGAT with GAT, it appears that pruning the original tree is effective.

The experimental results show that our model outperforms other neural network models and most graph-based models:

1) In the Lap14, Res15, MAMS, and Tweets datasets, the proposed model performed better than all baselines especially achieving increases of 0.73% ACC and 0.73% F1 on Tweets.

2) Compared with other datasets, there are more numbers and symbols in Lap14. The implicit syntactic relations that may be contained in these numbers and symbols are captured by our relational attention heads; therefore, our proposed model works well on Lap14.

3) In Res14, most of the reviews were directly about food, and there were fewer implicit sentiments in these reviews. Unlike other advanced models, this category of case was not our main target.

Table 4
The results of the case study in six sentences. "BiSyn-GAT" and "RGAT" are baselines and "DRGAT" is our model. "Pos", "Neu" and "Nes" respectively represent the predicted positive, neutral and negative sentiment polarities by the model. False predictions are marked with "✗" while true predictions are marked with "✓".

	Sentence	Aspect	BiSyn-GAT	RGAT	DRGAT
①	Enjoy using **Microsoft Office**!	**Microsoft Office**	Neu ✗	Pos ✓	Pos ✓
②	The only issue came when I tried **scanning** to the Mac.	**scanning**	Neu ✗	Nes ✓	Nes ✓
③	This laptop has only 2 **USB ports**, and they are both on the same side.	**USB ports**	Neu ✗	Neu ✗	Nes ✓
④	I work as a designer and coder and I needed a new buddy to work with, not **gaming**	**gaming**	Nes ✗	Nes ✗	Neu ✓
⑤	After fumbling around with the **OS** I started searching the internet for a fix and found a number of forums on fixing the issue.	**OS**	Nes ✓	Neu ✗	Nes ✓
⑥	Note,however, that any existing **MagSafe accessories** you have will not work with the **MagSafe 2 connection**.	**MagSafe accessories**	Neu ✓	Neg ✗	Neu ✓
		MagSafe 2 connection	Neu ✗	Neg ✓	Neg ✓
⑦	User upgradeable **RAM** and **HDD**.	**RAM**	Neu ✗	Pos ✓	Pos ✓
		HDD	Neu ✗	Neu ✗	Pos ✓
⑧	Also, in using the **built-in camera**, my **voice recording** for my vlog sounds like the interplanetary transmissions in the "Star Wars" saga.	**built-in camera**	Pos ✗	Neu ✗	Neg ✓
		voice recording	Pos ✗	Neu ✗	Neg ✓
⑨	I had the same reasons as most PC users: the **price**, the overbearing restrictions of **OSX** and lack of **support for games**.	**price**	Neu ✗	Neu ✗	Neg ✓
		OSX	Neg ✓	Neg ✓	Neg ✓
		support for games	Neg ✓	Neg ✓	Neg ✓
⑩	I bought it to my son who use it for **graphic design**.	**graphic design**	Neu ✓	Pos ✗	Neu ✓

4) In Res16, many sentences contained direct sentiment information. There are fewer sentences with implicit sentiment information and fewer neutral sentiments than other datasets. Our proposed model performs well with implicit sentiment information with syntactic meaning. However, it does not achieve the best result using this dataset.

5) Although our model does not consider the relations between multiple aspects; however, reviews have been conducted on MAMS regarding waiting times and waiter services. Our model captures the sentiment information using implicit syntactic relations. The result shows that the comments that are not directly related also contain implicit information which is helpful for sentiment classification.

6) All sentences in Tweets contain only one aspect, and the sentences are generally shorter than in the other datasets. Correspondingly, the height of our constituency tree was lower (fewer layers); therefore, the relations between words could be extracted more clearly. This may explain why the proposed model performs well in this case.

7) Compared with other datasets, the Tweets dataset contained more informal sentences, inverted sentences, and even grammatically incorrect sentences. Our model achieved the best performance on this dataset. One possible reason for this is that other models are more susceptible to interference when dealing with such sentences. Conversely, the relation attention mechanism captures the syntactic relations between words better than other models, enabling a deeper understanding of sentences and further improving the model's performance.

8) Another possible reason for the relatively poor performance of the proposed model on the Res14 dataset is the limited variety of relationships compared with other datasets. Compared to the other datasets, Rest14 has the fewest types of relations, which restricts the learning capacity of our model.

Overall, the proposed model performed well for complex sentences, sentences with implicit expressions, and sentences with a wide range of syntactic relations. However, it does not perform well when a high proportion of sentences express emotions directly.

4.3.2. Case study

The prediction results of the proposed model were compared with those of BiSyn-GAT [21] and RGAT [38], as shown in Table 4. Six sample examples demonstrated that the proposed model performed correctly in various situations. Through observation and analysis, we draw the following conclusions:

1) Sentences ① and ② can explain that when the sentence length is short, the height of the corresponding tree is generally low, and relational attention is used for sentiment classification better than a hierarchical structure.

2) Sentences ⑤ and ⑥ can show that hierarchical structures have the advantage of handling long sentences, but reconstruction and exploiting syntactic relations can greatly enhance the ability.

3) Regarding the comparison results of RGAT and DRGAT in sentences ③ and ④, we analyze that the dependencies may not be sufficient to provide complete syntactic information.

4) Sentence ⑥ proves our proposed model can also predict accurately in sentences with multiple aspect terms and complex structures.

Additionally, we have selected some sentences that do not directly express emotions to demonstrate our model's ability to handle ambiguous sentences:

1) In the sentence ⑦, the review merely states that both 'RAM' and 'HDD' are 'user upgradeable.' It does not explicitly convey any preference or aversion towards the laptop. This lack of clear emotional expression could potentially contribute to the confusion experienced by the other two models in their analysis.

2) Sentence ⑧, this sentence uses metaphor to indirectly express emotions, and even when two aspect words have different emotional polarities, our model is still able to make the correct judgment.

Table 5
Ablation results on six datasets. "Self heads" means multi-heads self attention, "con heads" denotes multi-heads constituency relational attention, and "dep heads" denotes multi-heads dependency relational attention. "α" means the value of RF-SF.

Models	Lap14		Res14		Res15		Res16		MAMS		Tweets	
	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1
W/ α =0.1	78.56	78.11	83.44	78.02	81.65	71.16	88.66	77.12	82.13	80.94	74.13	72.48
W/ α =0.4	80.95	78.66	83.12	78.32	84.58	69.25	89.36	78.13	82.62	81.61	77.85	77.12
W/ α =0.7	79.48	77.94	81.75	79.62	85.12	73.02	89.94	77.64	84.56	83.16	78.35	77.54
W/o reconstruction(α =1)	78.12	77.12	81.56	77.96	81.13	70.02	87.56	76.83	81.03	79.98	75.74	73.31
W/ structure A	80.12	79.52	83.66	77.61	84.02	71.98	90.69	77.65	84.02	84.56	76.64	75.91
W/ structure B	78.56	77.62	81.21	75.99	83.52	72.23	89.66	77.13	82.69	81.77	75.45	73.13
W/ structure C	77.66	74.12	80.66	75.21	82.13	70.68	87.52	76.13	83.68	82.35	77.56	76.23
W/ structure D	80.77	80.13	85.12	78.66	**85.47**	**73.24**	90.54	77.66	84.15	83.65	**78.84**	**77.73**
W/ structure E	**81.96**	**78.57**	**86.35**	**79.92**	84.35	72.66	**91.75**	**78.77**	**85.35**	**84.71**	78.21	77.43
W/o hierarchy	80.84	77.21	85.99	78.96	84.81	72.59	90.61	77.97	84.85	83.83	77.85	76.95
W/ only self heads	80.95	77.63	85.37	78.26	84.53	72.01	89.52	76.92	83.53	83.22	77.00	76.85
W/ only con heads	79.52	76.43	84.45	77.24	82.53	70.11	87.98	74.12	81.91	81.01	75.63	75.01
W/ only dep heads	80.12	77.62	85.11	78.49	84.12	71.19	88.65	75.55	82.88	81.73	76.02	75.63
W/o self heads	80.63	77.53	84.75	78.60	84.02	71.01	89.12	76.03	83.22	83.19	76.13	75.42
W/o con heads	81.45	78.13	85.85	79.61	85.29	73.11	90.99	78.01	84.63	84.02	77.84	77.08
W/o dep heads	81.06	77.92	85.57	79.36	85.01	72.89	90.53	77.66	84.83	84.11	77.01	76.58
DRGAT	**81.96**	**78.57**	**86.35**	**79.92**	**85.47**	**73.24**	**91.75**	**78.77**	**85.35**	**84.71**	**78.84**	**77.73**

3) In sentence ⑨, emotions are explicitly expressed only for the aspect terms "OSX" and "graphic design," while no sentiment is directly associated with the aspect term "price." Despite this complexity in the context, our model remains undisturbed and effectively processes the information.

4) Sentence ⑩ is a simple declarative sentence without any emotional inclination, and our model is capable of making the correct decision.

This case study shows that the syntactic relations between words and syntactic structure information can correct predictions, particularly when faced with complex sentences.

4.4. Ablation study

The main structure of the proposed model comprises four modules: Reconstruction, syntactic structure fusion, and a combination of hierarchy and different types of attention mechanisms. Different modules of the proposed model were considered to verify the unique advantages of each module.

- W/o reconstruction removes the reconstruction method based on RF-SF, which leverages the original constituency tree.
- W/ structures A-E indicate that our model chooses one syntactic structure from structure A to structure E.
- W/o hierarchy removes the hierarchical graphs. The graph corresponding to the layer where the aspect term is located in the constituency tree is the one used (i.e. layer 1 of the constituency tree).
- W/ only self heads, W/ only con heads and W/ only dep heads means using one type of multi-heads attention mechanism.
- W/o self heads, W/o con heads, and W/o dep heads eliminate one type of attention mechanism.

The ablation experimental results are presented in Table 5. The following conclusions are drawn:

1) W/o con heads performed the best when using two types of attention heads, which indicates that con heads provide the least help among the three types of heads.

2) Comparison of DRGAT with DRGAT W/o con heads shows that con heads are helpful for sentiment classification, although the effect is not significant.

3) From the results of W/o self-heads and W/ only self-heads, self-attention appears to be the most important. This indicates that full syntactic relational attention cannot replace self-attention.

4) From the W/o hierarchy, it can be proved that fine-grained to coarse-grained extracted features can greatly help sentiment classification.

5) W/ structure B shows that the dependency structure cannot replace the constituency structure and that stacking the dependency structure does not improve this effect. This is because the dependency structure contains noise.

6) The reason why W/ structure C is not satisfactory may be that the adjacency matrix is too sparse.

7) W/ structure D or W/ structure E achieve the best results in some of the datasets.

We conducted a series of experiments with four distinct values of α to ascertain its optimal value range. Considering the depth of the constituency trees and the number of relationship types found in sentences from the actual parsed dataset, we established a gradient of 0.3 for the differences between these varying α values. The following conclusions are drawn:

1) When the value of α is too high or too low the model's performance is not satisfactory. This indicates that retaining only a small fraction of the relation nodes, or even all, of them is not optimal.

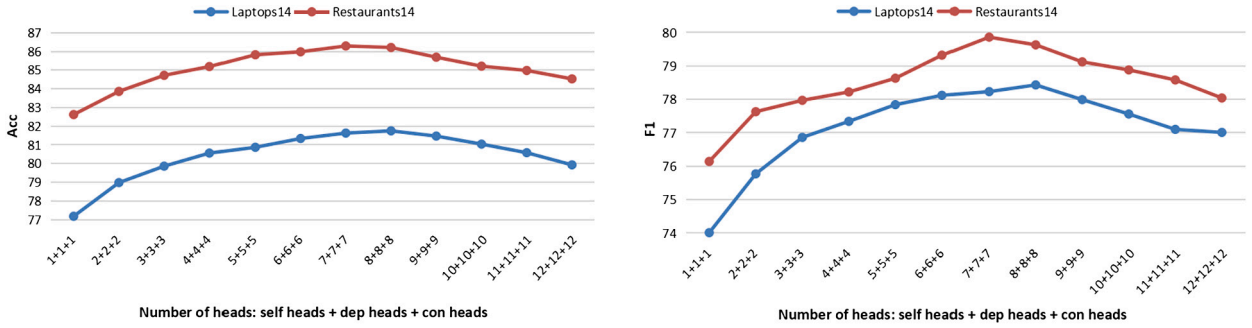


Fig. 5. Experimental results of the Accuracy and F1 of the number of attention heads.

Table 6
The Ablation of the hierarchy combination effect results. The "number" means the number of the hierarchy. "Self heads" means multi-heads self attention, "con heads" denotes multi-heads constituency relational attention, and "dep heads" denotes multi-heads dependency relational attention.

number	Lap14		Res14		Res15		Res16		MAMS		Tweets	
	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1
1	80.84	77.21	85.99	78.96	84.81	72.93	90.61	77.97	84.85	83.83	77.85	76.95
2	**81.96**	**78.57**	86.12	79.23	85.12	73.04	90.55	77.41	**85.35**	**84.71**	**78.84**	**77.73**
3	80.66	77.13	**86.35**	**79.92**	**85.47**	**73.21**	**91.75**	**78.77**	84.98	84.02	77.47	76.63
4	80.24	76.84	85.01	77.86	84.31	72.64	90.21	77.19	84.11	84.33	77.15	76.12

2) Compared to the other datasets, the Tweets dataset had the highest variety of constituency types. When α is set to 0.1, many important constituency relation nodes are discarded, which could be a reason why the performance of the model with α equal to 0.1 is worse than the model without reconstruction (α equal to 1).

3) Referring to the experimental results, the settings were adjusted to obtain the best results. In the Laptops 14 dataset, the value of α is set to 0.45, while in the Restaurants 14 dataset, it is set to 0.5. For the Restaurants 15, MAMS, and Tweets datasets, α is set at 0.65, and in the Restaurants 16 dataset, it is adjusted to 0.6.

When deleting edges between phrases of the dependency tree, some effective information might be deleted. This also implies that the method for choosing the phrase granularity in our model can be further improved.

4.5. Multi-relational-heads attention

The experiments are conducted with the datasets Lap14 and Res14 and with different heads of attention, as shown in Fig. 5 Through observation and analysis, the following conclusions can be made:

1) When the number of attention heads of each type is 1, the effect of the model is not satisfactory. This indicates that when the number of attention heads is unsatisfactory, it may not be sufficient to pay attention to the embedded information.

2) When the number of attention heads was 7-8, the proposed model achieved better results than the other models. When the number of each type of attention head was higher than 8, the ACC and F1 values of the model decreased with an increase in the number of heads. This is because when the number of heads increases to a certain extent, the model is overfitted.

3) Based on the results of this experiment, the settings were adjusted within in a small range. For example, in the Laptops 14 dataset, we set 8 self heads, 7 dep heads, and 6 con heads to achieve the best results.

4.6. Hierarchy combination effect

We introduced a mapping rule to select and combine representative graphs to form hierarchical graphs. To study the effectiveness of our mapping rules, several other methods for selecting and combining graphs were selected and cross-compared. First, we report the number of stacked DRGATs for which the model achieved the best performance.

The results are summarized in Table 6. The model achieved the best results when the number of DRGATs was in the range [2,3] on different datasets. It appears that when the number of layers is higher, the constituency tree cannot provide clear syntactic information. For example, as shown in Fig. 3, in the corresponding graph of the 4^{th} layer of the constituency tree, all the nodes are under one type of constituency relation. For different datasets, the number of DRGATs was set to either 2 or 3. After the number of DRGATs on each dataset was determined, several combinations were applied to verify the effectiveness of the proposed method.

- **Combination A** It is the way presented in the Syntactic Hierarchical Graphs Module of this article, i.e., Formula (7).
- **Combination B**

$$HA_B = \begin{cases} \{FA_j \mid j \in \{1,2,3,...,LS\}\} & if\ LS \le n \\ \{FA_1, FA_2, FA_3, ..., FA_n\} + \{FA_j \mid j \in \{n+1, n+2, n+3, ..., LS\}, FA_j = FA_1\} & if\ LS > n \end{cases} \tag{20}$$

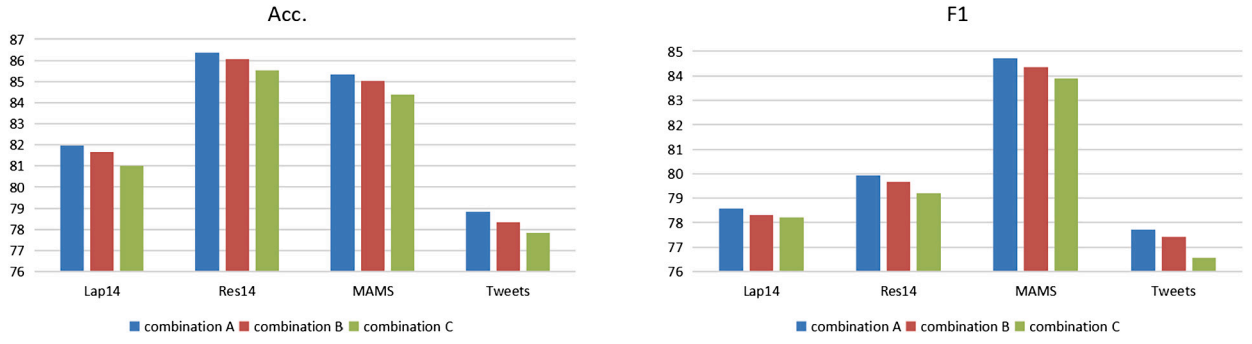


Fig. 6. Experimental results of combinations on four datasets.

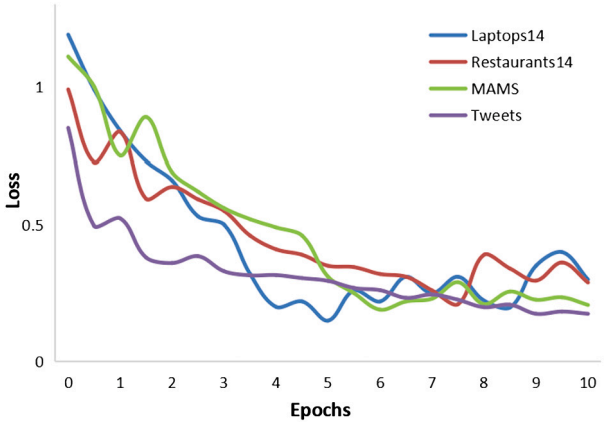


Fig. 7. The loss of DRGAT.

Compared with **Combination A**, this operation only considers that when $LS < n$, the corresponding graphs from layer 1 of the tree to the upper layer are selected. In other words, starting from the layer where the aspect words were located, the model exploited the syntactic structure of each adjacent layer upper in turn.

- **Combination C**

$$HA_C = \begin{cases} \{FA_j \mid j = n-i, i \in \{0,1,2,3,...,LS-1\}\} & if\ LS < n \\ \{FA_j \mid j \in \{1,2,3,...,LS\}\} & if\ LS = n \\ \{FA_1, FA_2, FA_3, ..., FA_n\} + \{FA_j \mid j \in \{n+1, n+2, n+3, ..., LS\}, FA_j = FA_1\} & if\ LS > n \end{cases} \tag{21}$$

Compared with **Combination A**, this operation only considers that when $LS < n$, the corresponding graphs are selected from the top layer of the tree to the layers below. In other words, starting from the top layer, the model exploited the syntactic structure of each adjacent layer below in turn.

We selected four datasets of Lap14, Res14, MAMS, and Tweets to carry out this experiment. The experimental results are shown in Fig. 6. It can be seen that:

1) Our combination and selection method performed better than the others.

2) The reason why combination A was better than combination B may be that in deep trees the syntactic information provided by adjacent layers is similar. Therefore, the interval-picking syntax graph is retained.

3) Combination C had the worst effect, indicating that some upper layers provided less value in syntactic information than the lower layers. In addition, the syntactic information of the layer where the aspect words are located was not obtained, which shows the importance of the aspect-specific layer.

4.7. Parameter sensitivity

Let us present the parameter variables that affect the model performance, and the experimental results are shown in Fig. 7 and Fig. 8.

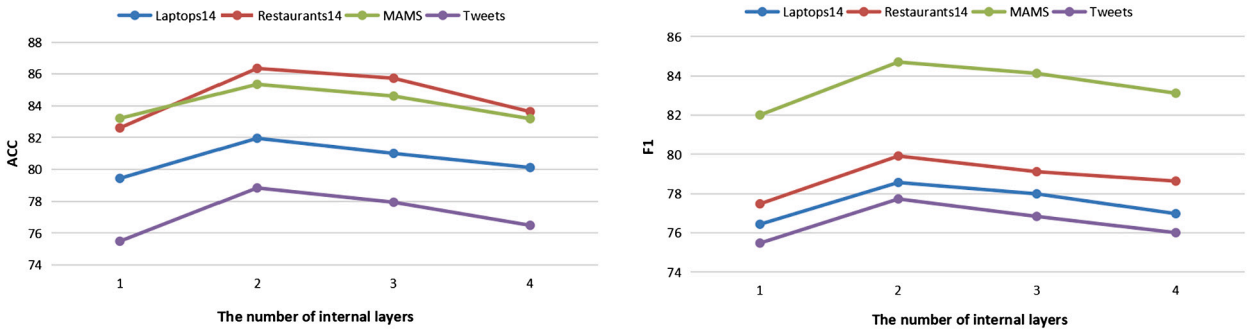


Fig. 8. Experimental results of the Accuracy and F1 of the number of internal layers.

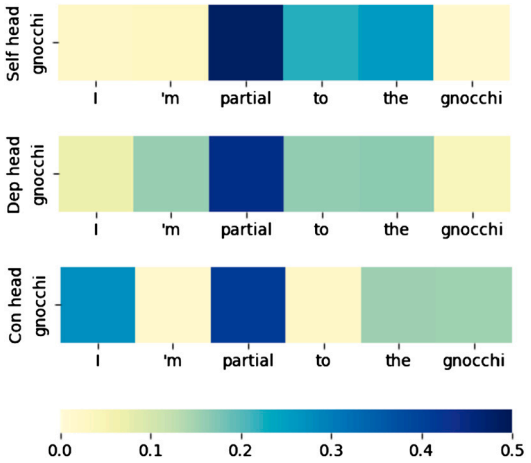


Fig. 9. An illustration of the visualization of three types of attention heads. The sample sentence "*I 'm partial to the gnocchi*" with an aspect "*gnocchi.*".

4.7.1. Loss of DRGAT

Fig. 7 shows that the proposed model gradually converged as the number of epochs increased. Without loss of generality, the experimental results are presented for four datasets: Laptops14, Restaurants14, MAMS, and Tweets. From the results, it can be observed that the different datasets converged at different numbers of epochs, with all datasets achieving convergence within 5-10 epochs. The Laptops dataset converged the fastest, reaching convergence at Epoch 5. The Tweets dataset had the slowest convergence speed, achieving convergence at the 10th Epoch. Thus, early stopping was introduced to train the model during the experiment.

4.7.2. DRGAT's internal layers: impact

To investigate the effect of the number of layers inside each DRGAT, we conducted experiments with different numbers of layers using the Laptops14, Restaurants14, MAMS, and Tweets datasets. The number of internal layers varied from 1 to 4 and the emerging performances are hereafter discussed.

From Fig. 8, it can be concluded that our model achieves the best performance when the number of internal layers is set to 2. Fig. 8 depicts the accuracy and F1 value across different numbers of layers. As the number of layers increases, the performance initially improves. However, after reaching 2 layers, the performance plateaued and even declined with additional layers.

This suggests that increasing the number of layers can enhance the representation capacity of a model by capturing more complex syntactic and relations between words.

4.8. Attention visualization

To investigate the different types of attention mechanisms, the aspect attention score matrix to all words of the sentence is visualized with different attention heads. An illustration of the attention score matrix when the three types of attention mechanisms are used independently is shown in Fig. 9. The following trends can be observed in the figure: 1) Different types of attention heads focus on different information, but they all pay the most attention to the "*partial*" that can provide sentiment information. This proves the effectiveness of these three types of attention mechanisms. 2) All three attention mechanisms pay attention to a part of redundant information; they do not pay more attention to "*partial.*" 3) The self-attention mechanism redundantly focuses on "*to*" and "*the,*" which may be that these two words are closer to "*gnocchi.*" 4) The redundant information of dependency relational attention is on "'*m,*" "*to,*" and "*the.*" These words are roots or have direct edges to "*gnocchi*" in the dependency tree. 5) In constituency relational attention, "*I*" does not provide emotional information, but is wrongly focused. This redundant information is limited by the structure.

6) When these attention mechanisms were used alone, they all focused on redundant information. As shown in Table 5, these kinds of attention do not work well when used alone.

4.9. Discussion for computational complexity

Similar to the operation of the self-attention layer, the operations of the two relational attentions can also be parallelized across all edges, and the computation of the output features can be parallelized across all nodes. The time complexity of a single attention head computing F_a features can be expressed as $O(|V|FF_a + |E|F_a)$, where F is the number of input features and $|V|$ and $|E|$ are the numbers of nodes and edges in the graph, respectively. As our model was stacked with several DRGATs, its time complexity was $O(L|V|FF_a + L|E|F_a)$, where L is the number of DRGATs.

Regarding the parsing dataset under consideration, constituency trees exhibit a significant number of layers, it is not uncommon to come across trees that have more than ten layers (i.e., L > 10). Under such circumstances, the number of DRGAT layers that we need to stack to process the corresponding graphs is also notably large. By reconstructing the constituency tree, we have effectively managed to limit the value of L to either 2 or 3, contingent upon the specific dataset in use. Therefore, the time complexity of our model is only a multiple of the difference compared to other baseline models and does not reach an exponential level.

Although our model employs K constituency heads, M dependency heads, and P self-attention heads, the computations within each head are entirely independent and can be parallelized. However, this significantly increases the storage requirements and the number of parameters.

5. Conclusion

This research introduces a hierarchical dual-relational ABSC graph attention network, whose peculiarity is to integrate syntactic structural and relational information. The proposed RF-SF method dynamically evaluates the importance of each relation node in each constituency tree. This provides a more concise and lower-depth tree structure by reconstructing the tree. The two types of relational attention heads added by DRGAT can provide a better understanding of implicit information embedded in sentences, particularly syntactic relational information. The model is more effective for sentences that contain implicit sentiment information, especially when the sentiment words are syntactically meaningful for the corresponding aspect words.

The proposed method has a few limitations. First, in cases where there are multiple aspect terms in a sentence, aspect-aspect relations are not considered, for simplicity. When a sentence contains multiple aspect terms, different positions in the syntactic structure may provide valuable information. Future work will apply syntactic relational attention mechanisms to model aspect-aspect relations. Furthermore, because of the need to employ three multi-head attention mechanisms simultaneously, our model carries a higher computational burden compared to the original GAT. The proposed model is currently not directly applicable to sentiment classification using other languages. However, if there is a syntactic parsing tool available for the target language and a corresponding pre-trained model, the proposed model can be applied. Ongoing work applies our model to Chinese datasets by leveraging Chinese-specific syntactic parsing tools and pre-trained models; however, its performance on Chinese datasets is yet to be evaluated.

CRediT authorship contribution statement

Lan You: Writing – review & editing, Supervision, Project administration, Conceptualization. **Jiaheng Peng:** Writing – original draft, Visualization, Validation, Software, Methodology, Formal analysis, Conceptualization. **Hong Jin:** Writing – review & editing, Project administration, Conceptualization. **Christophe Claramunt:** Writing – review & editing. **Haoqiu Zeng:** Resources, Data curation. **Zhen Zhang:** Data curation.

Declaration of competing interest

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.

Data availability

The data that has been used is confidential.

Acknowledgements

This work was partially supported by the Technology Innovation Special Program of Hubei Province (No. 2022BAA044, No. 2021BAA188), the Key Project of Science and Technology Research Program of Hubei Provincial Education Department (No. D20201006).

References

[1] Lingxian Bao, Patrik Lambert, Toni Badia, Attention and lexicon regularized lstm for aspect-based sentiment analysis, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, 2019, pp. 253–259.

[2] Chenhua Chen, Zhiyang Teng, Zhongqing Wang, Yue Zhang, Discrete opinion tree induction for aspect-based sentiment analysis, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2022, pp. 2051–2064.

[3] Peng Chen, Zhongqian Sun, Lidong Bing, Wei Yang, Recurrent attention network on memory for aspect sentiment analysis, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 452–461.

[4] Jiajun Cheng, Shenglin Zhao, Jiani Zhang, Irwin King, Xin Zhang, Hui Wang, Aspect-level sentiment classification with heat (hierarchical attention) network, in: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, 2017, pp. 97–106.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina 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), Minneapolis, Minnesota, June 2019, pp. 4171–4186, Association for Computational Linguistics.

[6] Xiaowen Ding, Bing Liu, The utility of linguistic rules in opinion mining, in: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2007, pp. 811–812.

[7] Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, Ke Xu, Adaptive recursive neural network for target-dependent Twitter sentiment classification, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2014, pp. 49–54.

[8] Timothy Dozat, Christopher D. Manning, Deep biaffine attention for neural dependency parsing, in: International Conference on Learning Representations, 2017.

[9] Binxuan Huang, Kathleen M. Carley, Parameterized convolutional neural networks for aspect level sentiment classification, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 1091–1096.

[10] Binxuan Huang, Yanglan Ou, Kathleen M. Carley, Aspect level sentiment classification with attention-over-attention neural networks, in: Social, Cultural, and Behavioral Modeling: 11th International Conference, SBP-BRiMS 2018, Washington, DC, USA, July 10-13, 2018, Proceedings 11, Springer, 2018, pp. 197–206.

[11] Lianzhe Huang, Xin Sun, Sujian Li, Linhao Zhang, Houfeng Wang, Syntax-aware graph attention network for aspect-level sentiment classification, in: Proceedings of the 28th International Conference on Computational Linguistics, 2020, pp. 799–810.

[12] Kentaro Inui, Jing Jiang, Vincent Ng, Xiaojun Wan, Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019.

[13] Qingnan Jiang, Lei Chen, Ruifeng Xu, Xiang Ao, Min Yang, A challenge dataset and effective models for aspect-based sentiment analysis, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 6280–6285.

[14] Di Jin, Zhizhi Yu, Pengfei Jiao, Shirui Pan, Dongxiao He, Jia Wu, Philip Yu, Weixiong Zhang, A survey of community detection approaches: from statistical modeling to deep learning, IEEE Trans. Knowl. Data Eng. (2021).

[15] D.K. Kirange, Ratnadeep R. Deshmukh, M.D.K. Kirange, Aspect based sentiment analysis semeval-2014 task 4, Asian J. Comp. Sci. Inf. Technol. 4 (2014).

[16] Himabindu Lakkaraju, Richard Socher, Chris Manning, Aspect specific sentiment analysis using hierarchical deep learning, in: NIPS Workshop on Deep Learning and Representation Learning, 2014, pp. 1–9.

[17] Ruifan Li, Hao Chen, Fangxiang Feng, Zhanyu Ma, Xiaojie Wang, Eduard 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.

[18] Yuncong Li, Cunxiang Yin, Sheng-hua Zhong, Sentence constituent-aware aspect-category sentiment analysis with graph attention networks, in: Natural Language Processing and Chinese Computing: 9th CCF International Conference, NLPCC 2020, Zhengzhou, China, October 14–18, 2020, Proceedings, Part I 9, Springer, 2020, pp. 815–827.

[19] Bin Liang, Hang Su, Lin Gui, Erik Cambria, Ruifeng Xu, Aspect-based sentiment analysis via affective knowledge enhanced graph convolutional networks, Knowl.-Based Syst. 235 (2022) 107643.

[20] Bin Liang, Rongdi Yin, Lin Gui, Jiachen Du, Ruifeng Xu, Jointly learning aspect-focused and inter-aspect relations with graph convolutional networks for aspect sentiment analysis, in: Proceedings of the 28th International Conference on Computational Linguistics, 2020, pp. 150–161.

[21] Shuo Liang, Wei Wei, Xian-Ling Mao, Fei Wang, Zhiyong He, Bisyn-gat+: Bi-syntax aware graph attention network for aspect-based sentiment analysis, in: Findings of the Association for Computational Linguistics: ACL 2022, 2022, pp. 1835–1848.

[22] Yuanyuan Liang, Yanbing Ju, Peiwu Dong, Xiao-Jun Zeng, Luis Martínez, Jinhua Dong, Aihua Wang, A sentiment analysis-based two-stage consensus model of large-scale group with core-periphery structure, Inf. Sci. 622 (2023) 808–841.

[23] Jiangming Liu, Yue Zhang, Attention modeling for targeted sentiment, in: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, 2017, pp. 572–577.

[24] Dehong Ma, Sujian Li, Xiaodong Zhang, Houfeng Wang, Interactive attention networks for aspect-level sentiment classification, in: Proceedings of the 26th International Joint Conference on Artificial Intelligence, 2017, pp. 4068–4074.

[25] Huyen Trang Phan, Ngoc Thanh Nguyen, Dosam Hwang, Convolutional attention neural network over graph structures for improving the performance of aspect-level sentiment analysis, Inf. Sci. 589 (2022) 416–439.

[26] Maria Pontiki, Dimitrios Galanis, Harris Papageorgiou, Suresh Manandhar, Ion Androutsopoulos, Semeval-2015 task 12: aspect based sentiment analysis, in: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), 2015, pp. 486–495.

[27] Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammed AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, et al., Semeval-2016 task 5: aspect based sentiment analysis, in: ProWorkshop on Semantic Evaluation (SemEval-2016), Association for Computational Linguistics, 2016, pp. 19–30.

[28] Guang Qiu, Bing Liu, Jiajun Bu, Chun Chen, Opinion word expansion and target extraction through double propagation, Comput. Linguist. 37 (1) (2011) 9–27.

[29] Youwei Song, Jiahai Wang, Tao Jiang, Zhiyue Liu, Yanghui Rao, Attentional encoder network for targeted sentiment classification, arXiv preprint, arXiv: 1902.09314, 2019.

[30] Mitchell Stern, Jacob Andreas, Dan Klein, A minimal span-based neural constituency parser, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2017, pp. 818–827.

[31] Duyu Tang, Bing Qin, Xiaocheng Feng, Ting Liu, Effective lstms for target-dependent sentiment classification, in: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, 2016, pp. 3298–3307.

[32] Duyu Tang, Bing Qin, Ting Liu, Aspect level sentiment classification with deep memory network, arXiv preprint, arXiv:1605.08900, 2016.

[33] Duyu Tang, Bing Qin, Ting Liu, Aspect level sentiment classification with deep memory network, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, Texas, Association for Computational Linguistics, November 2016, pp. 214–224.

[34] Hao Tang, Donghong Ji, Chenliang Li, Qiji Zhou, Dependency graph enhanced dual-transformer structure for aspect-based sentiment classification, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 6578–6588.

[35] Zemin Tang, Qi Xiao, Xu Zhou, Yangfan Li, Cen Chen, Kenli Li, Learning discriminative multi-relation representations for multimodal sentiment analysis, Inf. Sci. 641 (2023) 119125.

[36] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, Yoshua Bengio, Graph attention networks, arXiv preprint, arXiv:1710.10903, 2017.

[37] Jingjing Wang, Jie Li, Shoushan Li, Yangyang Kang, Min Zhang, Luo Si, Guodong Zhou, Aspect sentiment classification with both word-level and clause-level attention networks, in: IJCAI, vol. 2018, 2018, pp. 4439–4445.
[38] Kai Wang, Weizhou Shen, Yunyi Yang, Xiaojun Quan, Rui Wang, Relational graph attention network for aspect-based sentiment analysis, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 3229–3238.
[39] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, S. Yu Philip, A comprehensive survey on graph neural networks, IEEE Trans. Neural Netw. Learn. Syst. 32 (1) (2020) 4–24.
[40] Baiyu Yang, Donghong Han, Rui Zhou, Di Gao, Gang Wu, Aspect opinion routing network with interactive attention for aspect-based sentiment classification, Inf. Sci. 616 (2022) 52–65.
[41] Jiandian Zeng, Tianyi Liu, Weijia Jia, Jiantao Zhou, Relation construction for aspect-level sentiment classification, Inf. Sci. 586 (2022) 209–223.
[42] Chen Zhang, Qiuchi Li, Dawei Song, Aspect-based sentiment classification with aspect-specific graph convolutional networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 4568–4578.
[43] Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, Wai Lam, A survey on aspect-based sentiment analysis: tasks, methods, and challenges, IEEE Trans. Knowl. Data Eng. (2022).
[44] Yaojie Zhang, Bing Xu, Tiejun Zhao, Convolutional multi-head self-attention on memory for aspect sentiment classification, IEEE/CAA J. Autom. Sin. 7 (4) (2020) 1038–1044.
[45] Yu Zhang, Houquan Zhou, Zhenghua Li, Fast and accurate neural crf constituency parsing, in: Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, 2021, pp. 4046–4053.
[46] Meng Zhao, Jing Yang, Jianpei Zhang, Shenglong Wang, Aggregated graph convolutional networks for aspect-based sentiment classification, Inf. Sci. 600 (2022) 73–93.
[47] Pinlong Zhao, Linlin Hou, Ou Wu, Modeling sentiment dependencies with graph convolutional networks for aspect-level sentiment classification, Knowl.-Based Syst. 193 (2020) 105443.
[48] Yongqiang Zheng, Xia Li, Jian-Yun Nie, Store, share and transfer: learning and updating sentiment knowledge for aspect-based sentiment analysis, Inf. Sci. 635 (2023) 151–168.
[49] Junru Zhou, Zuchao Li, Hai Zhao, Parsing all: syntax and semantics, dependencies and spans, in: Findings of the Association for Computational Linguistics: EMNLP 2020, 2020, pp. 4438–4449.
[50] Tao Zhou, Kris Law, Douglas Creighton, A weakly-supervised graph-based joint sentiment topic model for multi-topic sentiment analysis, Inf. Sci. 609 (2022) 1030–1051.