Knowledge-Based Systems 236 (2022) 107736

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Phrase dependency relational graph attention network for Aspect-based Sentiment Analysis

Haiyan Wu [a,*], Zhiqiang Zhang [a], Shaoyun Shi [b], Qingfeng Wu [c], Haiyu Song [a]

[a] Zhejiang University of Finance and Economics, Hangzhou 310018, China
[b] Tsinghua University, Beijing 100084, China
[c] Chengdu Polytechnic, Chengdu 610095, China

ARTICLE INFO

Article history:
Received 4 June 2021
Received in revised form 9 November 2021
Accepted 9 November 2021
Available online 21 November 2021

Keywords:
Aspect-based Sentiment Analysis
Phrase dependency graph
Relational graph attention network

ABSTRACT

Aspect-based Sentiment Analysis (ABSA) is a subclass of sentiment analysis, which aims to identify the sentiment polarity such as positive, negative, or neutral for specific aspects or attributes that appear in a sentence. Previous studies have focused on extracting aspect-sentiment polarity pairs based on dependency trees, ignoring edge labels and phrase information. In this paper, we instead propose a phrase dependency graph attention network (PD-RGAT) on the ABSA task, which is a relational graph attention network constructed based on the phrase dependency graph, aggregating directed dependency edges and phrase information. We perform experiments with two pre-training models, GloVe and BERT. Experimental results on the benchmarking datasets (i.e., Twitter, Restaurant, and Laptop) demonstrate that our proposed PD-RGAT has comparable effectiveness to a range of state-of-the-art models and further illustrate that the graph convolutional structure based on the phrase dependency graph can capture both syntactic information and short long-range word dependencies. It also shows that incorporating directed edge labels and phrase information can enhance the capture of aspect-sentiment polarities on the ABSA task.

1. Introduction

Aspect-based Sentiment Analysis aims to distinguish the sentiment polarities (i.e, positive, negative, or neutral) over aspects terms in a sentence, belonging to a subtask of sentiment analysis [1]. However, sentiment analysis is a very important and challenging task in NLP. Indeed, the opportunity to automatically capture public sentiment about social events, political campaigns, marketing activities, and product preferences has attracted increasing attention from scientific and business communities. The former is an exciting open-ended challenge, while the latter is a remarkable result for marketing and financial market forecasting. This has led to the emergence of the emerging fields of sentiment computing and sentiment mining and analysis, which apply human-computer interaction, information retrieval, and multimodal signal processing to distill people's emotions from the growing amount of online social data [2,3]. For example, "although the menu is limited, all the vegetables are very fresh, and a work of gourmet art", the sentiment polarities of "menu" and "vegetables" are negative and positive, respectively. Therefore, in ABSA, the critical task is how to correctly mine the aspects terms, sentiment polarities, and the relationship between them in a sentence. Due to the complexity of natural language, except for analyzing the aspect terms and sentiment polarities containing in a sentence, it is also necessary to consider whether the sentence contains negative words, degree words, and other words that affect emotional expression.

On the ABSA task, many scholars have tried to employ different methods and have achieved some remarkable results. Early studies were feature-based methods, which are frequency-based feature engineering. Kiritchenko et al. first extracted high-frequency sentiment polarity words and aspect items separately, and then mined the aspect item-sentiment polarity pairs by clustering fusing some rules for the ABSA task [4]. Singh et al. utilized a SentiWordNet based scheme with adjectives, adverbs, and verbs and n-gram feature extraction to capture the aspect and semantic priority pairs in the sentence [5]. These methods are simple and effective. One obvious drawback of these frequency-based approaches is that not all common words refer to aspect-items. Some units of measurement, e.g., 'dollar' or 'ton' are frequently used, but they do not belong to aspects of sentiment polarities [6]. With the development of neural networks, some serialization models incorporating attention mechanisms have been proposed, e.g., ABCDM [7], AEC-LSTM [8], attention-based models (e.g., PF-CNN [9], AT-LSTM [10], AF-LSTM [11], ATAE-LSTM [12]) and

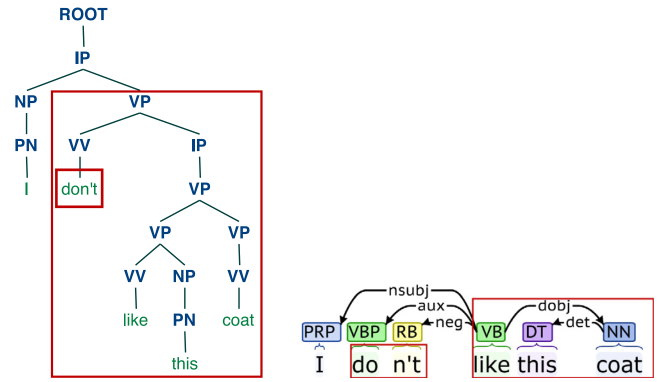
* Corresponding author.
E-mail addresses: wuhy2020@zufe.edu.cn (H. Wu), zqzhang@zufe.edu.cn (Z. Zhang), shisy17@mails.tsinghua.edu.cn (S. Shi), 11132019076@bnu.edu.cn (Q. Wu), hysong@zufe.edu.cn (H. Song).

https://doi.org/10.1016/j.knosys.2021.107736
0950-7051/

other combinatorial models like GCAE [13] and PosATT-LSTM [14]. These models mainly utilize serialized information as inputs based on character-level, word-level, POS, data fusion [15], etc. When an aspect term is separated from its sentiment polarity, finding the related semantic word in a sentence [16,17], some works focus on predicting sentiment intensity using stacked ensemble [18–20]. In addition, sentiments from these methods above are usually represented as positive, neutral, or negative. However, in many sentiment analysis problems, those neutral or contradictory comments are often ignored because of ambiguity and lack of information [21]. Some scholars conducted studies on this issue sincerely. For example, Valdivia et al. proposed using an induced ordered weighted average operator based on the fuzzy majority to aggregate several sentiment polarities [22]. Valdivia et al. also assigned neutrality by describing the boundary between positive and negative comments. The experimental results show that neutrality is the key to distinguishing between positive and negative and can effectively improve sentiment classification [23]. Wang et al. described a multi-level fine-scale sentiment analysis with ambivalence processing that can drill deeper into the text to reveal multi-level fine-scale sentiments and different types of emotions [24].

Recently, some studies utilized dependency trees to capture the aspect-sentiment polarity pairs [25]. For example, Ruder et al. demonstrated that "the knowledge of the review structure and sentential context should thus inform the classification of each sentence" is helpful by modeling a hierarchical bidirectional LSTM (Bi-LSTM) for ABSA tasks [26]. Zhao et al. proposed to utilize graph convolutional networks (GCNs) to capture the sentiment dependencies between multi-aspects in one sentence with dependency trees [27]. Dowlagar et al. presented the GCN with the multi-headed attention for sentiment analysis on the code-mixed text [28]. These studies also constructed graph neural networks (GNNs), graph attention networks (GAT) the gradual machine learning based on dependency trees for ABSA tasks [16,29,30]. To improve the effectiveness, some studies fused other information, e.g., knowledge information [31], edge information [32,33] into the dependency tree to construct GNNs for ABSA. Besides, some studies utilized phrase trees to build LSTMs, GCNs, or other models for ABSA tasks [34–36]. Comparing with those serialized models, the structured models based on phrase trees or dependency trees prove that applying syntactic information can better perform on the ABSA task, but they do not integrated both syntactic trees into the modeling. For dependency-based models, they neglect phrase information or dependency labels, and for phrase-based models, they neglect dependency information. In addition, some studies fuse both syntactic information to the models, but they do not make good use of the information of nodes or dependency edges.

Actually, two syntactic information, phrase structures and dependency relationships, are very helpful for the ABSA task. In linguistics, in terms of syntactic information itself, it plays a vital role in natural language. Syntactic information not only stipulates the composition rules between words but also contains the part of speech (POS) corresponding to each word. Phrase structure mines the bottom-to-high collocation relationship between sub-phrases or sub-trees in the sentence, whose importance has been demonstrated in previous studies [37,38]. Dependency tree explores the modification relationship between words in a sentence. The phrase structure and the dependency relationship, as two forms of syntactic structures, can both capture the collocation relationship between words in a sentence, where phrase tree explores the relationship between words in a sentence not limited to two words following a short collocation of word order, while dependency tree is mainly used to mine the modifying relationship between two words regardless of the distance of the



(a) An Example of Phrase Tree. (b) An Example of Dependency Tree.

Fig. 1. An example of a syntactic tree containing negative sentiment words.

word in a sentence. In other words, dependency trees are better at mining long-distance dependencies between words in a sentence. In contrast, phrase trees are good at exploring short-distance collocations between words in sentences, especially when they contain the negative word "not". Therefore, on the ABSA task, it is not sufficient to rely on the dependency tree to mine the aspect-sentiment polarity when negative sentiment words are involved. For example, the sentence "I don't like this coat", its phrase tree and dependency tree are shown in Fig. 1(a) and (b), respectively. Fig. 1(a) shows that the negative sentiment word "don't" belong to a whole VP phrase → VV IP ("don't like this coat") (shown in the red circle), while in Fig. 1(b), the negative word "don't" is not associated with "like this coat", which leads to the capture of the wrong sentiment polarity. Nevertheless, the dependency tree has a great advantage when the comments do not contain negative words. Therefore, we need to construct a phrase dependency graph that contains both a phrase tree and a dependency tree to capture the aspect-sentiment polarity pairs with a relational graph neural network that can capture the edge dependency label information.

To address the limitations of the aforementioned models ignoring the directed edge labels and phrase information, we take a further step in solving the short long-range word dependence problem by introducing phrase information and the directed dependency labels for the ABSA task. In this paper, we propose a **P**hrase **D**ependency **R**elational **G**raph **A**ttention **N**etwork (PD-RGAT) which fully exploits the advantage of pre-training and adequately models the syntactic relationship between different words with the phrase dependency graph. Specifically, PD-RGAT first obtains the representation of each word in the sentence from a global context bidirectional Long Short-Term Memory (Bi-LSTM) [39], which is proved to be very useful in mining word context related information [37]. Different from other Bi-LSTM based models which merge word representation to form word-level embeddings for a sentence, we retain all the representations of words and apply them in the subsequent phrase dependency graph to obtain more accurate representations of words by graph attention performed later. Then, we also construct a graph for each sentence according to its phrase dependency graphs. As we all know, in a syntactic tree, children nodes from the same parent node play different roles. Note that parameters of attention networks are related to category labels of syntactic nodes so that different modules have different attention parameters, and the same problem exists in dependency edges. To better model the children nodes in a phrase dependency graph, we construct the graph attention mechanism with multi-attention heads and multi-relation heads to pay different attentions to the children nodes and the dependency edges. Experiments conducted on three public datasets show that our proposed PD-RGAT model outstanding outperforms all baseline models on Twitter and Restaurant, and achieves more encouraging improvements. The main contributions are summarized as follows:

- We construct a phrase dependency graph with directed edge labels and phrase information by integrating the phrase tree and the dependency tree to explore both short and long dependencies between words in a sentence.
- A Phrase Dependency Relational Graph Attention Network (PD-RGAT) aggregating the node self information, neighboring node information, dependency information, and direct edge labels is proposed to model the phrase dependency graph with multi-attention node heads and multi-relation edge heads for the ABSA task.
- Experimental results on the benchmark datasets (i.e., Twitter, Restaurant and Laptop) demonstrate that our PD-RGAT significantly outperforms the state-of-the-art models on the ABSA task.

2. Related work

Sentiment analysis is a challenging research topic in natural language processing (NLP) [6]. In ABSA, the most important thing is how to effectively mine the "Aspect Item and Semantic Priority" relationship pairs [40]. The existing studies capturing the aspect item-sentiment polarities in a sentence are classified into three categories. • **Feature-based Methods**. These methods capture the aspect-items and sentiment polarity pairs by counting noun phrases or rule-based sentiment collocation phrases in the sentences, which are simple and effective and often suffer from not all high-frequency words associated with the aspect-items and semantic polarities [6]. • **Serialization-based Methods**. Some studies utilized LSTM or its variants (e.g., Bidirectional LSTM [41], Multilayer LSTM [42], LSTMs with attention mechanism [43–45], Hybrid LSTM [46]) to mine aspects and sentiment polarities in a sentence. These models somewhat alleviate the long dependence vanishing problem of RNN-based models [25,47]. Compared with frequency-based methods, such serialized methods mine the relationship between the aspect item and semantic polarity, ignoring the syntactic relationships between words in a sentence, especially the long-term word dependencies. Although the serialization model achieved improved results, the relational information of aspects in the same sentence was ignored, and the word location did not bring enough accurate information for ABSA. Some scholars have applied syntactic information, especially dependency relations, to perform the ABSA task. Lin et al. proposed a deep mask memory network with semantic dependency and context moment (DMMN-SDCM) integrating semantic parsing information of the aspect and the inter-aspect relation information into a deep memory network to perform the ABSA task [48]. In addition, some scholars have employed phrase information to structure the models. Wang et al. established a capsule tree-LSTM model which introduced a dynamic routing algorithm to build sentence representation by assigning different weights to nodes according to their contributions to predict the sentiment polarity [49]. From the experimental results, both the dependency relationship and phrase information can effectively capture the aspect item-sentiment polarity on the ABSA task. At present, few works employ the dependency relationship and phrase information together to build the structured model for the ABSA task. • **Structured-based Methods**. With the development of the Internet Network, graph data is becoming more and more common. Recently, the GCNs [50] or graph embeddings have attracted wide attention, especially GCNs based on textual information. Compared with previous studies, GCN-based models have been more effective on NLP tasks thought to have rich relational structure. They can capture the global structure information of a graph in graph embeddings [51]. However, using graph neural network methods, it is necessary to convert textual information into graph data, e.g., knowledge graphs, tree structures, or other structured data. In ABSA, building models based on the dependency tree has become a hot topic [1]. Some scholars have fused some features (e.g., sentiment words) into the dependency tree to construct GCN-based models. For example, Zhang et al. proposed an architecture based on hierarchical syntactic and lexical graphs to extract aspect-semantic polarity pairs [25]. Furthermore, some scholars applied knowledge graphs to mine the aspect item-sentiment polarity pairs. Fares et al. introduced an unsupervised word-level knowledge graph that uses different variants of the shortest path graph navigation technique to compute and propagate lexical knowledge bases (e.g., WordNet) to perform ABSA tasks [31].

Besides, some scholars make full use of graph information (e.g., node information) for modeling the graph structure. Huang et al. proposed a target-dependent graph attention network (TD-GAT) explicitly utilizing the dependency relationship among words for ABSA [1]. Although these GCN-based studies employ GCN or its variants to learn node representations from the dependency tree, ignoring edge label information. In NLP fields, scholars have achieved good results with relational graph neural networks constructed by fusing edge labeling information into GCN or GAT [52–54]. In the ABSA task, some studies integrated edge information to graph attention networks (GATs) [33]. Since the ASEGCN-based models aggregate the undirected edge label information, which is inconsistent with the directed edges of the dependency tree, the edge dependency information is weakened. To utilize the directed edges of the dependency tree, Jacket al. constructed the aspect-oriented dependency graph named Relational Graph Attention Network (RGAT) aggregating the directed edge information. The accuracy of RGAT has been significantly improved, which demonstrates that directed edges work better than undirected edges on ABSA tasks [55]. However, these advanced approaches generally ignore the phrase information which can identify the short-term range connections between aspects and opinion words.

Different from previous studies, this paper first constructs the phrase dependency graph with syntactic labels and syntactic information, integrating both the phrase structure and the dependency relationship. Secondly, we build RGAT using the phrase dependency graph to aggregate node self-information, neighboring node information, dependency information, and direct edge labels by utilizing multi-head and multi-head edge attention. The experimental results manifest that our proposed PD-RGAT model incorporating multiple information outperforms the state-of-the-art results on the ABSA task.

3. PD-RGAT: Phrase Dependency Relational Graph Attention Network

In this section, we introduce our PD-RGAT model. Firstly, we focus on how to construct the phrase dependency graph based on the phrase tree and dependency tree. Secondly, we describe in detail how to model phrase dependency graphs with R-GAT. Finally, we present the output of our PD-RGAT model and the associated training learning functions. The fluxogram overview of our PD-RGAT is shown in Fig. 2.

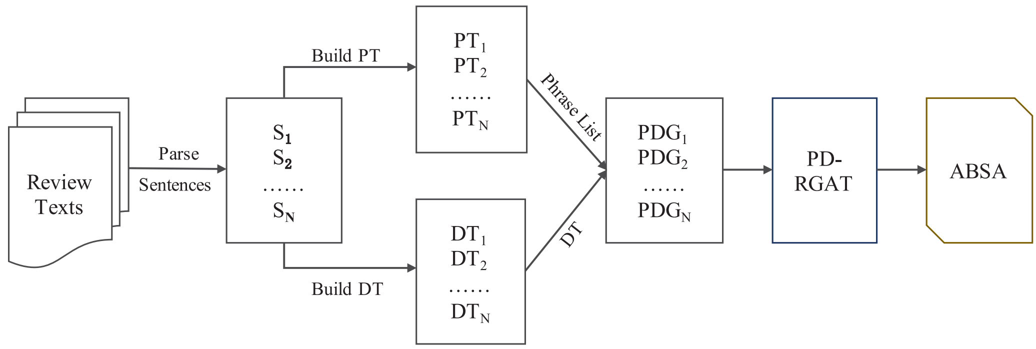


Fig. 2. A fluxogram overview of our PD-RGAT, where S_i refers to the i-th sentence, "PT" is "Phrase Tree", "DT" is "Dependency Tree", "PDG" is "Phrase Dependency Graph". Overall, the PD-RGAT model is divided into four steps: firstly, we parse the review texts into sentence sets. Secondly, we apply Stanford CoreNlP Toolkit to construct the review sentences' phrase tree and dependency tree. Then we capture the phrase list sets of the corresponding review sentences. Thirdly, the phrase set of review sentences and its corresponding dependency tree are utilized as the inputs of Algorithm 1 to build the phrase dependency graph (PDG). Finally, RGAT with PDG aggregates the node information, edge information, and syntactic dependency information on the edges respectively to perform the ABSA task.

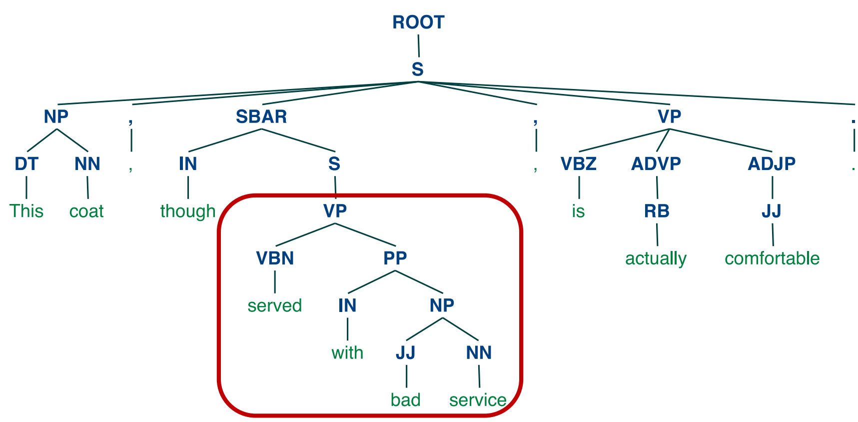


Fig. 3. An example of phrase tree capturing short-range dependencies between aspect item and sentiment polarity.

Algorithm 1: Phrase Dependency Graph

Input: dependency tree $T = (V, E)$, where $V = \left\{v_1, \cdots, v_{|V|}\right\}$, $E = \left\{\left(r_{ji}, v_i, v_j\right)\right\}$;
phrase list $P = \{p_1, \cdots, p_K\}$
Output: phrase dependency graph: $pT = (pV, pE)$, where $pV = \{T_1, \cdots, T_K\}$, $T_i = (V_i, E_i)$ and $pE = \left\{\left(r_{ji}, T_i, T_j\right)\right\}$

for *each phrase* $p_i \in P$ **do**
  $V_i \leftarrow \left\{v_j \mid v_j \in p_i\right\}$;
end
for *each edge* $(r_{nm}, v_m, v_n) \in E$ **do**
  $v_m \in p_k$, $v_n \in p_l$;
  if $k = l$ **then**
    $E_k \leftarrow E_k \cup \{(r_{nm}, v_m, v_n)\}$;
  end
  else
    $pE \leftarrow pE \cup \{(r_{nm}, T_k, T_l)\}$;
  end
end

3.1. Construction of phrase dependency graph

First, we employ the Stanford CoreNLP Toolkit [56] to construct the corresponding phrase trees and dependency trees for the input sentences. Taking "This coat, though served with bad service, is actually comfortable" as an example, its corresponding phrase tree and dependency tree are shown in Fig. 4. The steps for constructing a phrase dependency graph are as follows. Firstly, the phrases involved in the first-level child structure of the root node are extracted from the phrase tree corresponding to the sentence, as shown in Fig. 3, and the extracted phrase list P = NP[This coat], SBAR [though served with bad service], VP [is actually comfortable]. Secondly, we construct the dependency tree $T = (V, E)$, where T is the set of dependency tree nodes $V = \left\{v_1, \ldots, v_{|V|}\right\}$, $E = \left\{\left(r_{ji}, v_i, v_j\right)\right\}$, and E is the set of dependency tree edges. The list of phrases $P = \{p_1, \ldots, p_K\}$ as inputs to construct the phrase dependency graph $pT = (pV, pE)$, where $pV = \{T_1, \ldots, T_K\}$, $T_i = (V_i, E_i)$, $pE = \left\{\left(r_{ji}, T_i, T_j\right)\right\}$, and r_{ji} is the set of relations between two sub-graphs. According to Algorithm 1, the phrase dependency graph is constructed as shown in Fig. 5.

Although the phrase structure and the dependency relationship are two different kinds of syntactic information, they can be transformed and integrated each other [57–59]. Compared with dependency trees, phrase trees are more flexible and include more syntactic information, and the dependency information between words in a sentence is implicit in the phrase tree [60]. Therefore, a dependency tree incorporating phrase information will better capture the aspect-sentiment polarity pairs for ABSA. The converted and fused phrase dependency graph contains both phrase and dependency information, and the performance of the model constructed based on such structures has been greatly improved. In summary, since the task of this paper is aspect-based sentiment analysis, the phrase information is fused in the

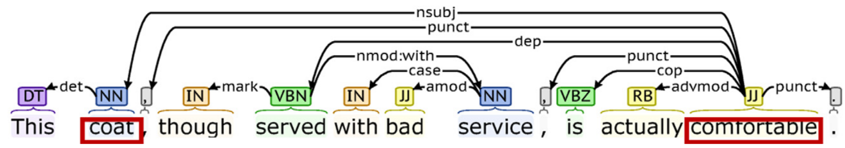


Fig. 4. An example of dependency tree capturing long-range dependencies between aspect item and sentiment polarity.

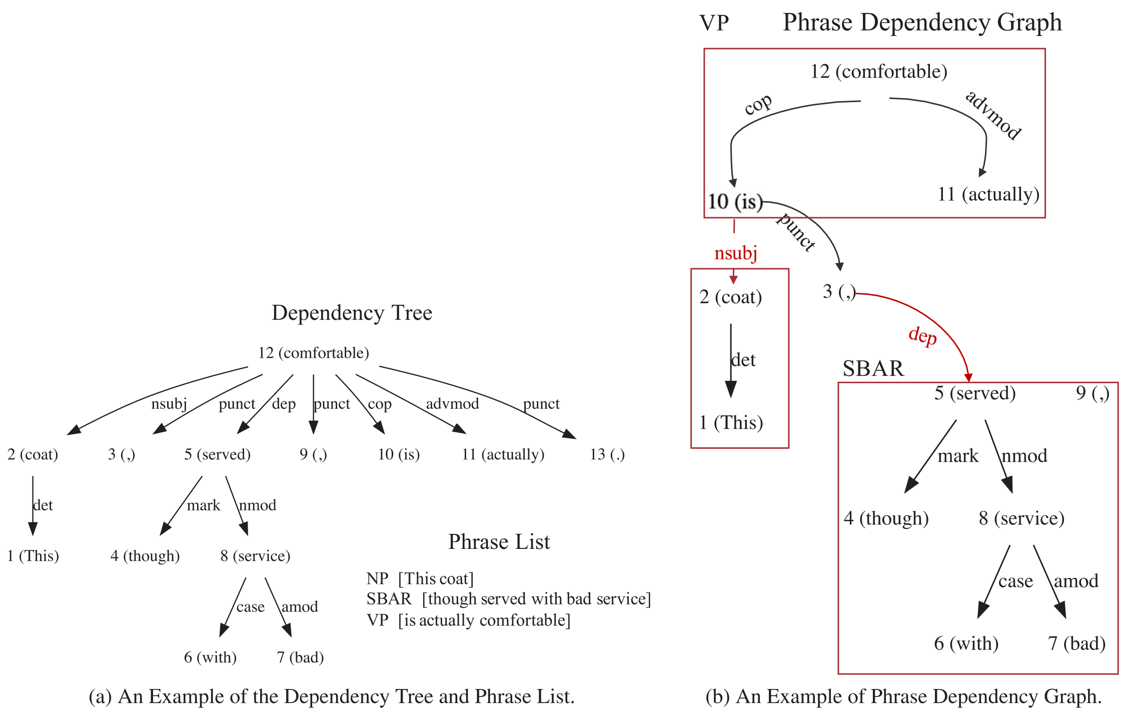


Fig. 5. An Example of the process of building a phrase dependency graph with Algorithm 1. *Left*: constructing the phrase tree and dependency tree of the example sentence ("this coat, though served with bad service, is actually comfortable".) employs Stanford CoreNLP Toolkit. Secondly, we obtain the example sentence's phrase list. *Right*: applying Algorithm 1 constructs the phrase dependency graph of the example sentence.

dependency tree while retaining the root node of the dependency tree, i.e., constructing a phrase dependency graph. This idea was proposed by Nguyen [56,61] and has been applied in other fields with remarkable results. Specifically, suppose a sentence "this coat, though served with bad service, is actually comfortable" often appears in comment related texts, and its corresponding phrase tree and dependency tree are shown in Figs. 3 and 4, respectively. In Fig. 3, the phrase tree mines the phrase "served with bad service" (shown in the red circle in Fig. 3) where the aspects term is "service" and the sentiment polarity is "bad". In Fig. 4, the dependency tree discovers the long-dependency modification relationship between words in a sentence, e.g., "comfortable coat" (shown in the red circle in Fig. 4). Algorithm 1 describes the above process. For an input sentence, we firstly apply Stanford CoreNLP Toolkit [62] to obtain its phrase tree and dependency tree, where r_{ij} is the dependency relationship from node i to j. Then we utilize Algorithm 1 to build the phrase dependency graph.

3.2. Modeling of phrase dependency graph

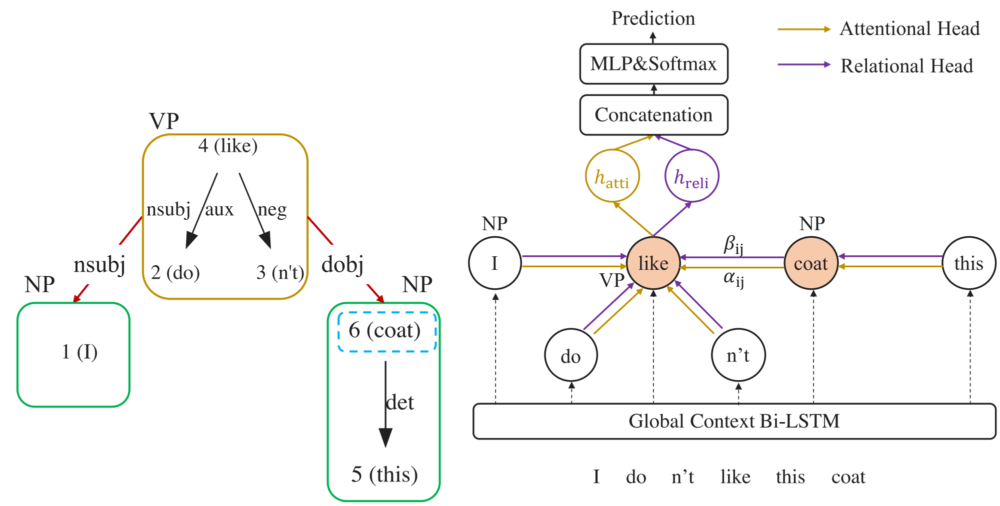
To explain how our PD-RGAT model works, we illustrate it with the simple sentence "I don't like this coat", whose phrase tree and dependency tree are shown in Fig. 1. The phrase dependency graph is constructed according to Algorithm 1. The phrase dependency graph is shown on the left side of Fig. 6, and the right side of Fig. 6 shows the overall framework of PD-RGAT.

3.2.1. Global context Bi-LSTM

In phrase dependency graph, the node represents the words of a sentence. For one sentence, single words contain no sequential information. Meanings of words can be inferred from their context. It is essential to represent words in a certain context. A bidirectional LSTM (Bi-LSTM) [39] is used to generate context-enhanced word vectors and global context vector. Suppose the input sentence s consists of a sequence of words $s = \{w_1, \dots, w_t, \dots, w_{|s|}\}$, where w_t is the tth word in the sentence and $|s|$ is the sentence length. We use bold fonts to represent the vectors of words and other objects. The word vectors $\mathbf{w}_t \in \mathbb{R}^d$ can be pre-trained vectors with GloVe [63] and BERT [64], respectively. and d is the dimension of word vectors. To enrich word vectors with the context information in the sentence, a Bi-LSTM is applied to the sequence of words $\{w_t\}_{t=1\dots|s|}$. Formally, the LSTM unit at position t is updated by:

$$
\begin{aligned}
\mathbf{i}_t &= \sigma(\mathbf{W}_i\mathbf{h}_{t-1} + \mathbf{U}_i\mathbf{w}_t + \mathbf{b}_i),\\
\mathbf{f}_t &= \sigma(\mathbf{W}_f\mathbf{h}_{t-1} + \mathbf{U}_f\mathbf{w}_t + \mathbf{b}_f),\\
\mathbf{g}_t &= tanh(\mathbf{W}_g\mathbf{h}_{t-1} + \mathbf{U}_g\mathbf{w}_t + \mathbf{b}_g),\\
\mathbf{o}_t &= \sigma(\mathbf{W}_o\mathbf{h}_{t-1} + \mathbf{U}_o\mathbf{w}_t + \mathbf{b}_o),\\
\mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t,\\
\mathbf{h}_t &= \mathbf{o}_t \odot tanh(\mathbf{c}_t)
\end{aligned}
\tag{1}
$$

where σ is the sigmoid function and $\odot$ is the element-wise product. $\mathbf{W}_i, \mathbf{W}_f, \mathbf{W}_g, \mathbf{W}_o \in \mathbb{R}^{d\times d}$ denote weight matrices of different gates for the hidden state $\mathbf{h}_t$, and $\mathbf{U}_i, \mathbf{U}_f, \mathbf{U}_g, \mathbf{U}_o \in \mathbb{R}^{d\times d}$ are weight matrices for the input $\mathbf{w}_t$. $\mathbf{b}_i, \mathbf{b}_f, \mathbf{b}_g, \mathbf{b}_o \in \mathbb{R}^d$ are bias



(a) An Example of Phrase Dependency Graph. (b) An Example of Building PD-RGAT based on PDG.

Fig. 6. An Example of the process of building PD-RGAT. *Left*: applying Algorithm 1 establishes the phrase dependency graph of the example sentence "I don't like this coat". *Rgith*: Building RGAT aggregates the node self information, neighboring node information, dependency information, and direct edge labels based on the phrase dependency graph with multi-attention node heads and multi-relation edge heads for ABSA.

vectors. The initial hidden states $\mathbf{h}_0$ can be zero initialized or other suitable vectors. Then, we apply another Bi-LSTM to encode the aspect words and utilize its average hidden state as the initial representation of the root.

3.2.2. RGAT: Relational Graph Attention Network

RGAT is an aggregation of the relationship information between nodes based on GAT. GAT operates on graph-structured data, leveraging masked self-attentional layers to address the shortcomings of prior methods based on graph convolutions or their approximations [65]. It is a variant of GNNs and has very powerful representation capabilities. GAT propagates information from an aspect's syntax context to the aspect node. Given a phrase dependency graphs G with N nodes, where each node can represent a word in the sentence and the edges of G denote the dependency between words. For an L-layer GAT network, features from L hops away can be propagated to the aspect target node to compute node representations by aggregating neighborhood's hidden states using multi-head attention:

$$h^{l+1}_{att_i} = \|_{k=1}^{K} \sum_{j\in\mathcal{N}_i} \alpha^{lk}_{ij} \mathbf{W}^{l}_{\mathbf{k}} \mathbf{h}^{l}_{\mathbf{j}} \tag{2}$$

$$\alpha^{lk}_{ij} = \frac{\exp\left(f\left(a^{T}_{ij}\left[\mathbf{W}_{\mathbf{ij}}\mathbf{h}^{\mathbf{l}}_{\mathbf{i}} \parallel \mathbf{W}_{\mathbf{ij}}\mathbf{h}^{\mathbf{k}}_{\mathbf{j}}\right]\right)\right)}{\sum_{u\in n[i]} \exp\left(f\left(a^{T}_{ij}\left[\mathbf{W}_{\mathbf{ij}}\mathbf{h}^{\mathbf{l}}_{\mathbf{i}} \parallel \mathbf{W}_{\mathbf{ij}}\mathbf{h}^{\mathbf{u}}_{\mathbf{i}}\right]\right)\right)} \tag{3}$$

where $\parallel$ represents vector concatenation, $h^{l+1}_{att_i}$ is the attention head of node i at layer l+ 1, α^{lk}_{ij} is the attention coefficient of node i to its neighbors $n[i]$ in attention head k at layer l. $W^{l}_{k} \in R^{\frac{D}{K}\times D}$ is an input transformation matrix for input states, where D is the dimension of hidden states, and σ is an activation function. $f(\cdot)$ denotes a LeakyReLU non-linear function [66].

Compared to GCN-based models, GAT mainly addresses how to determine the weight of each node to its different neighbors, which aggregates the representations of neighborhood nodes along the dependency paths. Most of the previous studies took no account of the directed dependency relationship between words considered the edge-dependent information but not the direction, which may lose some critical dependency information. In other words, neighboring nodes with different dependencies should have different impacts. To address this shortcoming, we add directed edge information between nodes to the original GAT model to represent dependency relationships between nodes, and then apply these relationship heads as relationship-wise gates to control information from neighboring nodes. This idea is inspired by the model RGAT [55]. Specifically, the dependency relationships are mapped to vector representations and then the relationship header update information is calculated as follows:

$$\mathbf{h}^{\mathbf{l+1}}_{\mathrm{rela}_i} = \|_{m=1}^{M} \sum_{j\in\mathcal{N}_i} \beta^{lm}_{ij} \mathbf{W}^{\mathbf{l}}_{\mathbf{m}} \mathbf{h}^{\mathbf{l}}_{\mathbf{j}} \tag{4}$$

$$g^{lm}_{ij} = \sigma\left(\mathrm{relu}\left(r_{ij}\mathbf{W}_{\mathbf{m}1} + \mathbf{b}_{\mathbf{m}1}\right)\mathbf{W}_{\mathbf{m}2} + \mathbf{b}_{\mathbf{m}2}\right) \tag{5}$$

$$\beta^{lm}_{ij} = \frac{\exp\left(g^{lm}_{ij}\right)}{\sum_{j=1}^{\mathcal{N}_i} \exp\left(g^{lm}_{ij}\right)} \tag{6}$$

where $h^{l+1}_{\mathrm{rela}_i}$ is the relational head of node i at layer l + 1, r_{ij} refers to the relation embedding between node i ans j. RGAT contains K multi-headed node attentions and M multi-headed edge relational attentions. Finally, each node i in the phrase dependency graph with N_i nodes is represented at layer l + 1 as follows:

$$\mathbf{x}^{\mathbf{l+1}}_{\mathbf{i}} = \mathbf{h}_{\mathbf{att}^{\mathbf{l}}_{\mathbf{i}}+1} \parallel \mathbf{h}_{\mathbf{rela}^{\mathbf{l}}_{\mathbf{i}}+1} \tag{7}$$

$$\mathbf{h}^{\mathbf{l+1}}_{\mathbf{i}} = \mathrm{relu}\left(\mathbf{W}_{\mathbf{l+1}}\mathbf{x}^{\mathbf{l+1}}_{\mathbf{i}} + \mathbf{b}_{\mathbf{l+1}}\right) \tag{8}$$

where $\parallel$ also represents vector concatenation, x^{l+1}_{i} is each node i associated with a local word embedding vector x at layer l + 1.

3.3. Output and learning

After applying Bi-LSTM and RGAT to the phrase dependency graph, the node vector containing contextual information is passed through RGAT. And its root node representation h^{l}_{a} can be mapped to probabilities over the different sentiment polarities through a fully connected softmax layer. The aggregation process goes from bottom to up, and finally, the root module outputs the sentence vector $\mathbf{m}_S$ for the downstream tasks (i.e., ABSA).

$$p(a) = \mathrm{softmax}\left(\mathbf{W}_{\mathbf{p}}\mathbf{h}^{l}_{\mathbf{a}} + \mathbf{b}_{\mathbf{p}}\right) \tag{9}$$

where p_a is the probability of the aspect a belonging to sentiment polarities.

Table 1
Statistics of evaluation datasets.

Dataset	Positive		Neural		Negative	
	Train	Test	Train	Test	Train	Test
Twitter	1561	173	3127	346	1560	173
Restaurant	2164	728	637	196	807	196
Laptop	994	341	464	169	870	128

The model is trained through *CrossEntropyLoss* [67], a well-known classification loss in machine learning:

$$L_\theta = -\sum_{(S,A)\in\mathcal{D}}\sum_{a\in A}\log p(a) \tag{10}$$

where A represents the aspects that appear in sentence S, D is all the aspect-sentiment pairs contained in sentence S, and θ is all the trainable parameters in the PD-RGAT model.

4. Experiment

4.1. DataSets

We evaluate the performance of our PD-RGAT on three benchmarking datasets: Twitter, Restaurant reviews and Laptop, where Restaurant reviews and Laptops are from the SemEval 2014 Task [68]. Twitter is one of the most popular social networking sites, allowing the users to read and post messages to express their opinions for the brands, celebrities, products and public events [69,70]. They are all utilized by the previous studies [1]. The specific statistical information of these datasets is shown in Table 1.

4.2. Parameters and implementation details

For a fair comparison of each model, we train all models with Adam [71] as the optimization function with the learning rate 10^{-3} to train our model. The weight of ℓ_2-regularization λ is manually searched between $\{0, 1\times10^{-5}, 1\times10^{-4}, 1\times10^{-3}, 1\times10^{-2}\}$. The learning rate is 1×10^{-3}, and early-stopping is conducted according to the performance on the validation set. Compared with the baseline models (as shown in Table 2), we apply two embedding methods (i.e., GloVe, BERT) in our paper. One is 300-dimensional GloVe embeddings [63], where we retrieve the corresponding embedding vector for each token in graphs and capture the relationship between each token (the directed edges between words). Another is BERT [64], with dimension 1024 implemented in PyTorch representations. The dimension of word vectors and hidden layers d is 300. We set the dimension dependency relation embeddings is 300. The dropout rate is 0.2. We run the experiments with five different random seeds and report the average accuracy and Macro-F1. All models are trained with two GPUs (NVIDIA GeForce GTX 2080Ti).

4.3. Baseline models

In particular, we compare our PD-RGAT model with very recent models on the benchmark datasets. The models we consider include:

- **Feature-based methods:**

(1) SVM utilizes n-gram features, parse features and lexicon features (2014) for aspect-based sentiment classification [4].

- **Attention-based methods**:

(1) Attention-based Long Short-Term Memory Network, **ATAE-LSTM (2016)** [12] employs the attention mechanism concentrating on different parts of a sentence when various aspects are taken as inputs.
(2) Interactive Attention Networks, **IAN (2017)** [44] can interactively learn attentions in the contexts and targets and generate the representations for targets and contexts separately.
(3) Recurrent Attention on Memory, **RAM (2017)** [45] adopts the multiple-attention mechanism to capture sentiment features separated by a long distance so that it is more robust against irrelevant information.
(4) Position-aware Bidirectional Attention Network, **PBAN (2018)** [41] not only concentrates on the position information of aspect terms but also mutually models the relation between aspect term and sentence by employing a bidirectional attention mechanism.
(5) Multi-Grained Attention Network, **MGAN (2018)** [72] applies a fine-grained attention mechanism that can capture the word-level interaction between aspect and context.
(6) Target-Dependent LSTM, **TD-LSTM (2015)** [73] integrates the connections between the target word and context words when building a learning system.
(7) Attention-based LSTM, **AT-LSTM (2016)** [10] utilizes attention mechanism and LSTM to capture the importance of each context word towards a target by modeling their semantic associations.
(8) Deep Memory Network, **MemNet (2016)** [74] employs a deep memory network to capture the importance of each context word when inferring the sentiment polarity of an aspect.

- **Syntax-aware methods:**

(1) Adaptive Recursive Neural Network, **AdaRNN (2014)** [69] adaptively propagates the sentiments of words to target depending on the context and syntactic relationships between them.
(2) Phrase Recursive Neural Network, **PhraseRNN (2015)** [61] is an extension of RNN (Recursive Neural Network) based on dependency and constituent trees of a sentence into account.
(3) Attention-based LSTM, **LSTM+SynATT (2018)** [75] utilizes LSTM for target representation and then app an attention model incorporating syntactic information into the attention mechanism.
(4) Aspect-Specific Graph Convolutional Networks, **ASGCN (2019)** [76] builds a Graph Convolutional Network (GCN) over the dependency tree of a sentence to exploit syntactical information and word dependencies.
(5) Syntactic Edge-Enhanced Graph Convolutional Networks, **ASEGCN (2020)** [33] can effectively learn better representations of aspects and the opinion words by considering the different types of neighborhoods with the edge constraint.
(6) Transformation Networks, **TNet (2018)** [77] employs a CNN layer instead of attention to extract salient features from the transformed word representations originated from a bi-directional RNN layer.
(7) Convolution Over a Dependency Tree, **CDT (2019)** [29] exploits a Bi-LSTM to learn representations for features of a sentence, and then enhances the embeddings with a graph convolutional network, which operates directly on the dependency tree of the sentence.
(8) Syntax and Knowledge via Graph Convolutional Network, **SK-GCN (2020)** [32] leverages the syntactic dependency tree and commonsense knowledge via GCN by modeling the syntactic dependency tree and commonsense knowledge graph.

Table 2
Overall performance on three benchmark datasets.

Models	Twitter		Restaurant		Laptop	
	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
Feature+SVM (2014)	–	–	80.2	–	70.5	–
AdaRNN (2014)	66.30	65.90	–	–	–	–
PhraseRNN (2015)	–	–	66.20	59.32	–	–
LSTM+SynATT (2018)	–	–	80.45	71.26	72.57	69.13
ASGCN (2019)	72.15	70.40	80.77	72.02	75.55	71.05
ASEGCN (2020)	75.96	73.98	81.31	78.47	84.43	76.59
TNet (2018)	73.12	71.01	80.79	70.84	76.54	71.75
CDT (2019)	74.66	73.66	82.30	74.02	77.19	72.99
SK-GCN (2020)	75.00	73.01	83.48	75.19	79.00	75.57
SD-GCN (2020)	–	–	83.57	76.47	81.35	78.34
BiGCN (2020)	74.16	73.35	81.97	73.48	74.59	71.84
GAT (2017)	71.67	70.13	78.21	67.17	73.04	68.11
TD-GAT (2019)	72.68	71.15	80.35	76.13	74.13	72.01
R-GAT (2020)	75.51	73.80	83.29	77.33	73.74	75.56
GGCN-SR (2020)	–	–	87.20	82.5	**82.8**	**80.2**
ATAE-LSTM (2016)	–	–	77.20	–	68.70	–
IAN (2017)	–	–	78.60	–	72.10	–
RAM (2017)	69.36	67.30	80.23	70.80	74.49	71.35
PBAN (2018)	–	–	81.16	–	74.12	–
MGAN (2018)	72.54	70.81	81.25	71.94	75.39	72.47
TD-LSTM (2015)	70.80	69.00	78.00	68.43	71.83	68.43
AT-LSTM (2016)	73.51	71.93	81.63	71.94	77.43	73.70
MemNet (2016)	69.65	67.68	78.16	65.83	70.33	64.09
GCAE (2018)	–	–	77.28	–	69.14	–
JCI (2018)	–	–	–	68.84	–	67.23
TNET (2018)	74.90	73.60	80.69	71.27	76.54	71.75
Ours: PD-RGAT (GloVe)	76.90	75.23	88.01	83.26	81.07	80.15
Ours: PD-RGAT (BERT)	**77.86**	**76.21**	**88.67**	**83.55**	81.63	80.87

(9) Sentiment Dependencies Graph Convolutional Networks, **SD-GCN (2020)** [78] employ a bidirectional attention mechanism with position encoding to model aspect-specific representations between each aspect and its context words. And then uses GCN over the attention mechanism captures the sentiment dependencies between different aspects in one sentence.

(10) Hierarchical Syntactic and Lexical Graphs, **BiGCN (2020)** [25] builds a concept hierarchy on both the syntactic and lexical graphs for differentiating various types of dependency relations or lexical word pairs.

(11) Graph Attention Network, **GAT (2017)** [65] leverages masked self-attentional layers to address the shortcomings of prior methods based on graph convolutions or their approximations.

(12) Target-Dependent Graph Attention Network, **TD-GAT (2019)** [79] explicitly utilizes the dependency relationship among words and propagates sentiment features directly from the syntactic context of an aspect target based on the dependency graph.

(13) Relational Graph Attention Network, **R-GAT (2020)** [55] defines a unified aspect-oriented dependency tree structure rooted at a target aspect by reshaping and pruning an ordinary dependency parse tree, and then apply a relational graph attention network to encode the new tree structure for sentiment prediction.

(14) Gated Graph Convolutional Networks and Syntax-based Regulation, **GGCN-SR (2020)** [80] utilizes gate vectors to customize the hidden vectors of the graph-based models toward the aspect terms and uses dependency trees to obtain the importance scores for each word in the sentences.

- **Other recent methods:**

(1) Gated Convolutional network with Aspect Embedding, **GCAE (2018)** [13] is a more accurate and efficient model based on convolutional neural networks and gating mechanisms.

(2) Target Sensitive Memory Networks with Joint Coupled Interaction, **JCI (2018)** [81] extracts the sentiment polarity of the (detected) context depending on the given target and cannot be inferred from the context alone by the joint coupled interaction.

(3) Transformation Networks, **TNET (2018)** [82] employs a CNN layer to extract salient features from the transformed word representations originated from a bi-directional RNN layer.

4.4. Experimental results and analysis

The overall performances are shown in Table 2, from which several observations can be concluded. Firstly, the PD-RGAT model outperforms most of the baseline models on Twitter and Restaurant, and the GGCN-SR model performs on Laptop. Secondly, the performance of R-GAT can be significantly improved when incorporated with relational heads and phrase structures in our phrase dependency tree. PD-RGAT also outperforms most of the syntax-aware models (e.g., TD-GAT, GAT, ASGCN, BiGCN, SD-GCN, SK-GCN, CDT, TNet, ASGCN, LSTM+SynATT, PhraseRNN, and AdaRNN), which also involve syntactic information building different models, attention-based methods (e.g., ATAE-LSTM, IAN, Feature-based methods RAM, PBAM, MGAN, TD-LSTM, AT-LSTM, MemNet, HSCN and CapsNet) and feature-based methods. This proves that the proposed model PD-RGAT is better at encoding the phrase and dependency information. Thirdly, R-GAT constructed based on Phrase Dependency tree on three benchmarking datasets is superior to R-GAT based on Dependency tree, which shows that phrase information is also crucial on the ABSA task. Fourthly, in terms of TD-LSTM and TD-GAT, they are both based on dependency trees construct models. Still, the TD-GAT model outperforms TD-LSTM, which demonstrates the superiority of graph neural networks in information captured and representation, which also illustrates the superiority of RGAT used in our model PD-RGAT. In addition, for PD-RGAT itself, two pre-trained

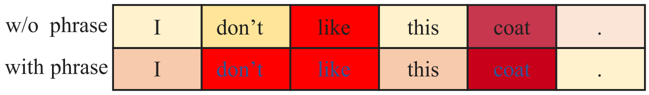


Fig. 7. An example of attention weights obtained by PD-RGAT with phrase and without phrase respectively.

Table 3
Results of ablation study of PD-RGAT.

Model	Twitter	Restaurant	Laptop
PD-RGAT (BERT)	**77.86**	**88.67**	**81.63**
w/o global RNN	77.85	88.65	81.60
w/o phrase label	77.83	88.59	81.58
w/o edge information	77.75	87.97	81.01

models, GloVe and BERT, are used, where the BERT-based models are more accurate than the Glove-based ones, demonstrating the power of this pre-trained model on the ABSA task. However, after integrating our PD-RGAT (R-GAT+BERT), this strong model has achieved a new state-of-the-art. These results demonstrate the effectiveness of our PD-RGAT in capturing important phrase structures, dependency information, and edge labels for the ABSA task.

4.5. Ablation and case study

Two studies, Ablation Study and Case Study, were done separately to verify the effectiveness of the PD-RGAT model and its components proposed in this paper. Among them, Ablation Study analyzed the importance of each component of the PD-RGAT model through accuracy. In contrast, Case Study examined the impact of each component of the PD-RGAT model on the attention scores of the constituents of the sentences during the training process from specific examples.

4.5.1. Ablation study

To investigate the ablation of different parts, we removed the global RNN, phrase label, and edge information from the PD-RGAT model. The overall performances of the ablation study are shown in Table 3, in which "w/o" means "without". The results show that all these parts contribute to the excellent performance of PD-RGAT. The global RNN enriches the word vector with contextual information and provides a global contextual representation of the sentence. The phrase label captures the syntactic relationships between nodes in the upper and lower layers while showing the role of words and phrases in the sentence. Guided by edge information, PD-RGAT dynamically aggregates information from child nodes and edge dependency relationships.

4.5.2. Case study

To verify the impact of phrase structure and directed edge information on our PD-RGAT model, we apply two examples as a case study. We utilize heatmaps to visualize the attention values of the words computed by our proposed PD-RGAT model in the two examples, as shown in Figs. 7 and 8, where the darker the color, the more the model pays attention to it.

The first sentence "I don't like this coat" shows that PD-RGAT containing phrasal information can capture the negative sentiment word "don't like" for aspect "coat" well. In contrast, without phrasal information, PD-RGAT does not extract the negative sentiment verb for the aspect "coat", and only mine the sentiment verb "like", which is the opposite of the meaning of the original aspect word "coat".

The second example sentence, "this coat, though served with bad service, is actually comfortable" is mainly used to illustrate that directed edge information has better results in mining long dependencies between words in a sentence. Fig. 8 manifests that PD-RGAT with directed edge label information can accurately capture the aspect-sentiment polarity pairs in the sentence, even though there are not close, e.g., the green "coat comfortable" and red "bad service" in the second sentence of Fig. 8. However, PD-RGAT does not contain information about the directed edge label information, which capture the wrong aspect-sentiment polarity pairs with close location distance, e.g., "coat …bad" and "service …comfortable" as shown in the first sentence of Fig. 8.

5. Conclusion

In this work, we propose a Phrase Dependency Relational Graph Attention Network, named PD-RGAT, which applies RGAT to model the phrase dependency graph by aggregating the information of nodes, edges, and phrase information. Specifically, we build a phrase dependency graph with the edge labels and phrase information integrating phrase tree and dependency tree for a whole corpus. We firstly apply global context information to represent words in a sentence. The graph convolution layer improves the representation of sub-graphs by aggregating nodes, phrase information, and edge labels. Experimental results on three public datasets show that the connections between aspects and opinion words in a sentence can be better captured by PD-RGAT based on the phrase dependency graph. The performance of RGAT is significantly improved compared to state-of-the-art methods on the ABSA task. Our future work will conduct experiments on the phrase dependency relational graph attention network and more NLP tasks. Besides, we will integrate the sentiment of neutrality or ambivalence as an enhanced part of the ABSA task into the phrase dependency graph attention network and experiment on more NLP tasks.

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.

Acknowledgments

This work was partly supported by the National Key Research and Development Program of China (No. 2019YFB1406300), the National Natural Science Foundation of China under Grant (18ZDA238), the National Natural Science Foundation of China under Grant (61972336, 62073284) and Zhejiang Provincial Natural Science Foundation of China under Grant (LY22F020027, LY19F030008).

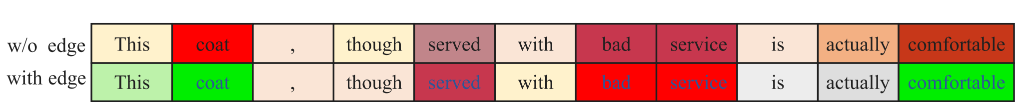


Fig. 8. An example of attention weights obtained by PD-RGAT with edge labels and without edge labels respectively.

References

[1] L. Huang, X. Sun, S. Li, L. Zhang, H. Wang, Syntax-aware graph attention network for aspect-level sentiment classification, in: COLING, International Committee on Computational Linguistics, 2020, pp. 799–810.

[2] E. Cambria, D. Das, S. Bandyopadhyay, A. Feraco, Affective computing and sentiment analysis, in: A Practical Guide to Sentiment Analysis, Springer, 2017, pp. 1–10.

[3] A. Hussain, E. Cambria, S. Poria, A.Y.A. Hawalah, F. Herrera, Information fusion for affective computing and sentiment analysis, Inf. Fusion 71 (2021) 97–98.

[4] S. Kiritchenko, X. Zhu, C. Cherry, S. Mohammad, NRC-Canada-2014: Detecting aspects and sentiment in customer reviews, in: SemEval@COLING, The Association for Computer Linguistics, 2014, pp. 437–442.

[5] V.K. Singh, R. Piryani, A. Uddin, P. Waila, Sentiment analysis of movie reviews: A new feature-based heuristic for aspect-level sentiment classification, in: 2013 International Mutli-Conference on Automation, Computing, Communication, Control and Compressed Sensing (IMac4s), IEEE, 2013, pp. 712–717.

[6] K. Schouten, F. Frasincar, Survey on aspect-level sentiment analysis, IEEE Trans. Knowl. Data Eng. 28 (3) (2016) 813–830.

[7] M.E. Basiri, S. Nemati, M. Abdar, E. Cambria, U.R. Acharya, ABCDM: An attention-based bidirectional CNN-RNN deep model for sentiment analysis, Future Gener. Comput. Syst. 115 (2021) 279–294.

[8] F. Huang, X. Li, C. Yuan, S. Zhang, J. Zhang, S. Qiao, Attention-emotion-enhanced convolutional LSTM for sentiment analysis, IEEE Trans. Neural Netw. Learn. Syst. (2021).

[9] B. Huang, K.M. Carley, Parameterized convolutional neural networks for aspect level sentiment classification, in: EMNLP, Association for Computational Linguistics, 2018, pp. 1091–1096.

[10] Y. Wang, M. Huang, X. Zhu, L. Zhao, Attention-based LSTM for aspect-level sentiment classification, in: EMNLP, The Association for Computational Linguistics, 2016, pp. 606–615.

[11] Y. Tay, L.A. Tuan, S.C. Hui, Learning to attend via word-aspect associative fusion for aspect-based sentiment analysis, in: AAAI, AAAI Press, 2018, pp. 5956–5963.

[12] 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, 2016, pp. 606–615.

[13] W. Xue, T. Li, Aspect based sentiment analysis with gated convolutional networks, in: ACL (1), Association for Computational Linguistics, 2018, pp. 2514–2523.

[14] J. Zeng, X. Ma, K. Zhou, Enhancing attention-based LSTM with position context for aspect-level sentiment classification, IEEE Access 7 (2019) 20462–20471.

[15] F. Chen, Z. Yuan, Y. Huang, Multi-source data fusion for aspect-level sentiment classification, Knowl. Based Syst. 187 (2020).

[16] B. Huang, K.M. Carley, Syntax-aware aspect level sentiment classification with graph attention networks, in: EMNLP/IJCNLP (1), Association for Computational Linguistics, 2019, pp. 5468–5476.

[17] S. Al-Azani, E.-S.M. El-Alfy, Using word embedding and ensemble learning for highly imbalanced data sentiment analysis in short arabic text, Procedia Comput. Sci. 109 (2017) 359–366.

[18] Y. Zhang, J. Du, X. Ma, H. Wen, G. Fortino, Aspect-based sentiment analysis for user reviews, Cogn. Comput. (2021) 1–14.

[19] M.E. Basiri, S. Nemati, M. Abdar, S. Asadi, U.R. Acharrya, A novel fusion-based deep learning model for sentiment analysis of COVID-19 tweets, Knowl.-Based Syst. 228 (2021) 107242.

[20] J. Zhang, A. Zhang, D. Liu, Y. Bian, Customer preferences extraction for air purifiers based on fine-grained sentiment analysis of online reviews, Knowl.-Based Syst. 228 (2021) 107259.

[21] L.C. Hand, B.D. Ching, Maintaining neutrality: A sentiment analysis of police agency facebook pages before and after a fatal officer-involved shooting of a citizen, Gov. Inf. Q. 37 (1) (2020).

[22] A. Valdivia, M.V. Luzíón, F. Herrera, Neutrality in the sentiment analysis problem based on fuzzy majority, in: 2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), IEEE, 2017, pp. 1–6.

[23] A. Valdivia, M.V. Luzón, E. Cambria, F. Herrera, Consensus vote models for detecting and filtering neutrality in sentiment analysis, Inf. Fusion 44 (2018) 126–135.

[24] Z. Wang, S. Ho, E. Cambria, Multi-level fine-scaled sentiment sensing with ambivalence handling, Int. J. Uncertain. Fuzziness Knowl. Based Syst. 28 (4) (2020) 683–697.

[25] M. Zhang, T. Qian, Convolution over hierarchical syntactic and lexical graphs for aspect level sentiment analysis, in: EMNLP (1), Association for Computational Linguistics, 2020, pp. 3540–3549.

[26] S. Ruder, P. Ghaffari, J.G. Breslin, A hierarchical model of reviews for aspect-based sentiment analysis, in: EMNLP, The Association for Computational Linguistics, 2016, pp. 999–1005.

[27] P. Zhao, L. Hou, O. Wu, Modeling sentiment dependencies with graph convolutional networks for aspect-level sentiment classification, Knowl. Based Syst. 193 (2020) 105443.

[28] S. Dowlagar, R. Mamidi, Graph convolutional networks with multi-headed attention for code-mixed sentiment analysis, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, 2021, pp. 65–72.

[29] K. Sun, R. Zhang, S. Mensah, Y. Mao, X. Liu, Aspect-level sentiment analysis via convolution over dependency tree, in: EMNLP/IJCNLP (1), Association for Computational Linguistics, 2019, pp. 5678–5687.

[30] Y. Wang, Q. Chen, J. Shen, B. Hou, M.H.M. Ahmed, Z. Li, Aspect-level sentiment analysis based on gradual machine learning, Knowl. Based Syst. 212 (2021) 106509.

[31] M. Fares, A. Moufarrej, E. Jreij, J. Tekli, W.I. Grosky, Unsupervised word-level affect analysis and propagation in a lexical knowledge graph, Knowl. Based Syst. 165 (2019) 432–459.

[32] J. Zhou, J.X. Huang, Q.V. Hu, L. He, SK-GCN: modeling syntax and knowledge via graph convolutional network for aspect-level sentiment classification, Knowl. Based Syst. 205 (2020) 106292.

[33] Y. Xiao, G. Zhou, Syntactic edge-enhanced graph convolutional networks for aspect-level sentiment classification with interactive attention, IEEE Access 8 (2020) 157068–157080.

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

[35] B. Wang, W. Lu, Learning latent opinions for aspect-level sentiment classification, in: AAAI, AAAI Press, 2018, pp. 5537–5544.

[36] A. Sahay, A. Maheshwari, R. Kumar, G. Ramakrishnan, M.K. Hanawal, K. Arya, Unsupervised learning of explainable parse trees for improved generalisation, 2021, ArXiv e-prints, arXiv–2104.

[37] H. Wu, Y. Liu, S. Shi, Modularized syntactic neural networks for sentence classification, in: EMNLP (1), Association for Computational Linguistics, 2020, pp. 2786–2792.

[38] H. Shi, H. Zhou, J. Chen, L. Li, On tree-based neural sentence modeling, in: EMNLP, Association for Computational Linguistics, 2018, pp. 4631–4641.

[39] M. Schuster, K.K. Paliwal, Bidirectional recurrent neural networks, IEEE Trans. Signal Process. 45 (11) (1997) 2673–2681.

[40] X. Guo, W. Yu, X. Wang, An overview on fine-grained text sentiment analysis: Survey and challenges, in: J. Phys. Conf. Ser., 1757, (1) IOP Publishing, 2021, 012038.

[41] S. Gu, L. Zhang, Y. Hou, Y. Song, A position-aware bidirectional attention network for aspect-level sentiment analysis, in: COLING, Association for Computational Linguistics, 2018, pp. 774–784.

[42] H. Sun, T. Jiang, Y. Dai, Sentiment analysis of commodity reviews based on multilayer LSTM network, in: Proceedings of the International Conference on Artificial Intelligence, Information Processing and Cloud Computing, 2019, pp. 1–5.

[43] S. Rida-E-Fatima, A. Javed, A. Banjar, A. Irtaza, H. Dawood, H. Dawood, A. Alamri, A multi-layer dual attention deep learning model with refined word embeddings for aspect-based sentiment analysis, IEEE Access 7 (2019) 114795–114807.

[44] D. Ma, S. Li, X. Zhang, H. Wang, Interactive attention networks for aspect-level sentiment classification, 2017, arXiv preprint arXiv:1709.00893.

[45] P. Chen, Z. Sun, L. Bing, W. Yang, Recurrent attention network on memory for aspect sentiment analysis, in: EMNLP, Association for Computational Linguistics, 2017, pp. 452–461.

[46] N. Chen, P. Wang, Advanced combined lstm-cnn model for twitter sentiment analysis, in: 2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS), IEEE, 2018, pp. 684–687.

[47] Y. Ji, H. Liu, B. He, X. Xiao, H. Wu, Y. Yu, Diversified multiple instance learning for document-level multi-aspect sentiment classification, in: EMNLP (1), Association for Computational Linguistics, 2020, pp. 7012–7023.

[48] K. Sun, R. Zhang, S. Mensah, Y. Mao, X. Liu, Aspect-level sentiment analysis via convolution over dependency tree, 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. 5679–5688.

[49] S. Gu, L. Zhang, Y. Hou, Y. Song, A position-aware bidirectional attention network for aspect-level sentiment analysis, in: Proceedings of the 27th International Conference on Computational Linguistics, 2018, pp. 774–784.

[50] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, 2016, arXiv preprint arXiv:1609.02907.

[51] L. Yao, C. Mao, Y. Luo, Graph convolutional networks for text classification, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 01, 2019, pp. 7370–7377.

[52] D. Busbridge, D. Sherburn, P. Cavallo, N.Y. Hammerla, Relational graph attention networks, 2019, CoRR, arXiv:1904.05811.

[53] Z. Zhang, F. Zhuang, H. Zhu, Z. Shi, H. Xiong, Q. He, Relational graph neural network with hierarchical attention for knowledge graph completion, in: AAAI, AAAI Press, 2020, pp. 9612–9619.

[54] M.S. Schlichtkrull, T.N. Kipf, P. Bloem, R. van den Berg, I. Titov, M. Welling, Modeling relational data with graph convolutional networks, 2017, CoRR, arXiv:1703.06103.
[55] K. Wang, W. Shen, Y. Yang, X. Quan, R. Wang, Relational graph attention network for aspect-based sentiment analysis, in: ACL, Association for Computational Linguistics, 2020, pp. 3229–3238.
[56] R. Delmonte, Treebanking in VIT: from phrase structure to dependency representation, in: Language Engineering for Lesser-Studied Languages, IOS Press, Amsterdam, the Netherlands, 2009, pp. 51–80.
[57] Z. Wang, C. Zong, Phrase structure parsing with dependency structure, in: COLING (Posters), Chinese Information Processing Society of China, 2010, pp. 1292–1300.
[58] M. de Marneffe, B. MacCartney, C.D. Manning, Generating typed dependency parses from phrase structure parses, in: LREC, European Language Resources Association (ELRA), 2006, pp. 449–454.
[59] F. Xia, M. Palmer, Converting dependency structures to phrase structures, Tech. rep., Pennsylvania Univ Philadelphia, 2001.
[60] F. Xia, M. Palmer, Converting dependency structures to phrase structures, in: HLT, Morgan Kaufmann, 2001.
[61] T.H. Nguyen, K. Shirai, PhraseRNN: Phrase recursive neural network for aspect-based sentiment analysis, in: EMNLP, The Association for Computational Linguistics, 2015, pp. 2509–2514.
[62] C.D. Manning, M. Surdeanu, J. Bauer, J.R. Finkel, S. Bethard, D. McClosky, The stanford corenlp natural language processing toolkit, in: ACL (System Demonstrations), The Association for Computer Linguistics, 2014, pp. 55–60.
[63] J. Pennington, R. Socher, C.D. Manning, Glove: Global vectors for word representation, in: EMNLP, ACL, 2014, pp. 1532–1543.
[64] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, 2018, CoRR, arXiv:1810.04805.
[65] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph attention networks, 2017, CoRR, arXiv:1710.10903.
[66] A.L. Maas, A.Y. Hannun, A.Y. Ng, Rectifier nonlinearities improve neural network acoustic models, in: Proc. Icml, 30, (1) Citeseer, 2013, p. 3.
[67] P.-T. De Boer, D.P. Kroese, S. Mannor, R.Y. Rubinstein, A tutorial on the cross-entropy method, Ann. Oper. Res. 134 (1) (2005) 19–67.
[68] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, Semeval-2014 task 4: Aspect based sentiment analysis, in: SemEval@COLING, The Association for Computer Linguistics, 2014, pp. 27–35.
[69] L. Dong, F. Wei, C. Tan, D. Tang, M. Zhou, K. 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.
[70] H.H. Do, P.W.C. Prasad, A. Maag, A. Alsadoon, Deep learning for aspect-based sentiment analysis: A comparative review, Expert Syst. Appl. 118 (2019) 272–299.
[71] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.
[72] F. Fan, Y. Feng, D. Zhao, Multi-grained attention network for aspect-level sentiment classification, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 3433–3442.
[73] D. Tang, B. Qin, X. Feng, T. Liu, Effective LSTMs for target-dependent sentiment classification, 2015, arXiv preprint arXiv:1512.01100.
[74] D. Tang, B. Qin, T. Liu, Aspect level sentiment classification with deep memory network, in: EMNLP, The Association for Computational Linguistics, 2016, pp. 214–224.
[75] S. He, Z. Li, H. Zhao, H. Bai, Syntax for semantic role labeling, to be, or not to be, in: ACL (1), Association for Computational Linguistics, 2018, pp. 2061–2071.
[76] C. Zhang, Q. Li, D. Song, Aspect-based sentiment classification with aspect-specific graph convolutional networks, 2019, arXiv preprint arXiv:1909.03477.
[77] X. Li, L. Bing, W. Lam, B. Shi, Transformation networks for target-oriented sentiment classification, 2018, arXiv preprint arXiv:1805.01086.
[78] P. Zhao, L. Hou, O. Wu, Modeling sentiment dependencies with graph convolutional networks for aspect-level sentiment classification, Knowl.-Based Syst. 193 (2020) 105443.
[79] B. Huang, K.M. Carley, Syntax-aware aspect level sentiment classification with graph attention networks, 2019, arXiv preprint arXiv:1909.02606.
[80] A.P.B. Veyseh, N. Nouri, F. Dernoncourt, Q.H. Tran, D. Dou, T.H. Nguyen, Improving aspect-based sentiment analysis with gated graph convolutional networks and syntax-based regulation, in: EMNLP (Findings), Association for Computational Linguistics, 2020, pp. 4543–4548.
[81] S. Wang, S. Mazumder, B. Liu, M. Zhou, Y. Chang, Target-sensitive memory networks for aspect sentiment classification, in: ACL (1), Association for Computational Linguistics, 2018, pp. 957–967.
[82] X. Li, L. Bing, W. Lam, B. Shi, Transformation networks for target-oriented sentiment classification, in: ACL (1), Association for Computational Linguistics, 2018, pp. 946–956.