BiSyn-GAT+: Bi-Syntax Aware Graph Attention Network for Aspect-based Sentiment Analysis

Shuo Liang [1] **, Wei Wei** [2,✉]**, Xian-Ling Mao** [3]**, Fei Wang** [4]**, Zhiyong He** [5]

[1,2] Cognitive Computing and Intelligent Information Processing (CCIIP) Laboratory,
School of Computer Science and Technology, Huazhong University of Science and Technology
[3] School of Computer Science and Technology, Beijing Institute of Technology
[4] Institute of Computig Technology, Chinese Academy of Sciences
[5] Naval University of Engineering
[1] shuoliang@hust.edu.cn, [2] weiw@hust.edu.cn, [3] maoxl@bit.edu.cn,
[4] wangfei@ict.ac.cn, [5] moonmon_pub@outlook.com

Abstract

Aspect-based sentiment analysis (ABSA) is a fine-grained sentiment analysis task that aims to align aspects and corresponding sentiments for aspect-specific sentiment polarity inference. It is challenging because a sentence may contain multiple aspects or complicated (*e.g.,* conditional, coordinating, or adversative) relations. Recently, exploiting dependency syntax information with graph neural networks has been the most popular trend. Despite its success, methods that heavily rely on the dependency tree pose challenges in accurately modeling the alignment of the aspects and their words indicative of sentiment, since the dependency tree may provide noisy signals of unrelated associations (*e.g.,* the "*conj*" relation between "*great*" and "*dreadful*" in Figure 2). In this paper, to alleviate this problem, we propose a **Bi-Syn**tax aware **G**raph **At**tention Network (**BiSyn-GAT+**). Specifically, **BiSyn-GAT+** fully exploits the syntax information (*e.g.,* phrase segmentation and hierarchical structure) of the constituent tree of a sentence to model the sentiment-aware context of every single aspect (called ***intra***-context) and the sentiment relations across aspects (called ***inter***-context) for learning. Experiments on four benchmark datasets demonstrate that BiSyn-GAT+ outperforms the state-of-the-art methods consistently.

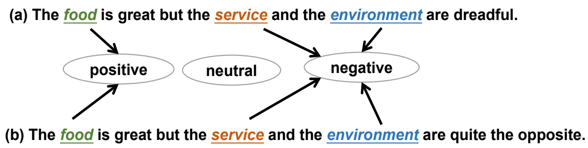

Figure 1: Examples of ABSA task. Each underlined aspect is classified to corresponding sentiment polarity.

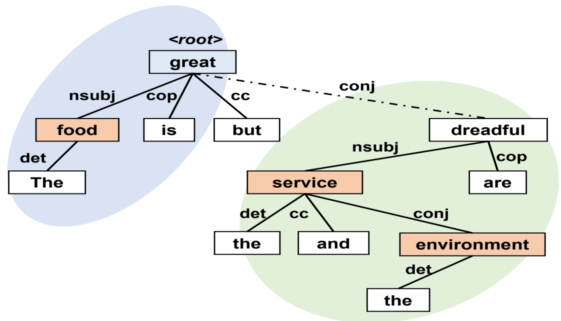

Figure 2: Dependency tree of "The food is great but the service and the environment are dreadful". Two separate ellipses encircle its two clauses. The "conj" edge between "great" and "dreadful" is a noise.

1 Introduction

Aspect-based sentiment analysis (ABSA) aims to identify the sentiment polarity towards a given aspect in the sentence. Many previous works (Yang et al., 2018; Li et al., 2019) mainly focus on extracting sequence features via Recurrent Neural Networks (RNNs) or Convolution Neural Networks (CNNs) with attention mechanisms, which often assume that words closer to the target aspect are more likely to be related to its sentiment. However, the assumption might not be valid as exemplified in Figure 1 (a), "service" is obviously closer to "great" rather than "dreadful", and these methods may assign the irrelevant opinion word "great" to "service" mistakenly.

To mitigate this problem, there already exists several efforts (Wang et al., 2020a; Chen et al., 2020) dedicated to research on how to effectively leverage non-sequential information (*e.g.,* syntactic information like dependency tree) via Graph Neural Networks (GNNs). Generally, a dependency tree (*i.e.,* Dep.Tree), linking the aspect terms to the syntactically related words, stays valid in the long-distance dependency problem. However, the inherent nature of Dep.Tree structure may introduce noise like the unrelated relations across clauses, such as "conj" relation between "great" and "dreadful" in Figure 2, which discourages capturing the sentiment-aware context of each aspect, *i.e., intra*-context. More-

✉ Corresponding Author

Findings of the Association for Computational Linguistics: ACL 2022, pages 1835 - 1848
May 22-27, 2022

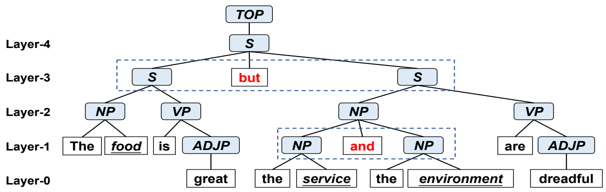

Figure 3: Constituent tree of the sentence "The food is great but the service and the environment are dreadful". Context words are in rectangles and parsed phrase types are in rounded rectangles.

over, the Dep.Tree structure only reveals relations between words and, thereby, in most cases, is incapable of modeling complicated (*e.g.,* conditional, coordinating, or adversative) relations of sentences, therefore failing to capture sentiment relations between aspects, *i.e., inter*-context.

Hence, in this paper, we consider fully exploiting the syntax information of the constituent tree to tackle the problem. Typically, a constituent tree (*i.e.,* Con.Tree) often contains precise and discriminative phrase segmentation and hierarchical composition structure, which are helpful for correctly aligning the aspects and their corresponding words indicative of sentiment. The former can naturally divide a complicated sentence into multiple clauses, and the latter can discriminate different relations among aspects to infer the sentiment relations of different aspects. We illustrate this with an example in Figure 3: (1) Clause "The food is great" and the clause "the service and environment are dreadful" are segmented by the phrase segmentation term "but"; (2) In Layer-1, the term "and" indicates the coordinating relation of "service" and "environment", while the term "but" in Layer-3 reflects the adversative relation towards "food" and "service" (or "environment").

Thus, to better align aspect terms and corresponding sentiments, we propose a new framework, **Bi**-**Syn**tax aware **G**raph **At**tention Network (**BiSyn-GAT+**), to effectively leverage the syntax information of constituent tree by modeling *intra*-context and *inter*-context information. In particular, BiSyn-GAT+ employs: 1) a syntax graph embedding to encode the *intra*-context of each aspect based on the fusion syntax information within the same clause in a bottom-up way, which combines the phrase-level syntax information of its constituent tree and the clause-level syntax information of its dependency tree. 2) an aspect-context graph consisting of phrase segmentation terms and all aspects to model the *inter*-context of each aspect. Specifically, it aggregates the sentiment information of other aspects according to the influence between the current aspect and its neighbor aspects, which is calculated based on aspect representations learned from bi-directional relations over the aspect context graph, respectively.

Our main contributions are as follows:

(1) To the best of our knowledge, this is the first work to exploit syntax information of constituent tree (*e.g.,* phrase segmentation and hierarchical structure) with GNNs for ABSA. Moreover, it shows superiority in the alignments between aspects and corresponding words indicative of sentiment.

(2) We propose a framework, **Bi**-**Syn**tax aware **G**raph **At**tention Network (**BiSyn-GAT+**), to fully leverage syntax information of constituent tree (or, and dependency tree) by modeling the sentiment-aware context of each single aspect and the sentiment relations across aspects.

(3) Extensive experiments on four datasets show that our proposed model achieves state-of-the-art performances.

2 Related Work

Sentiment analysis is an important task in the field of natural language processing (Zhang et al., 2018; Yang et al., 2020) and can be applied in downstream tasks, like emotional chatbot (Wei et al., 2019; Li et al., 2020a; Lan et al., 2020; Wei et al., 2021), recommendation system (Zhao et al., 2022; Wang et al., 2020b), QA system (Wei et al., 2011; Qiu et al., 2021). Here we focus on a fine-grained sentiment analysis task — ABSA. Recently, deep learning methods have been widely adopted for ABSA task. These works can be divided into two main categories: methods without syntax information (*i.e.,* Syntax-free methods) and methods with syntax information (*i.e.,* Syntax-based methods).

Syntax-free methods: Neural networks with attention mechanisms (Wang et al., 2016; Chen et al., 2017; Song et al., 2019) have been widely used. Chen et al. (2017) adopts a multiple-attention mechanism to capture sentiment features. Song et al. (2019) uses an attentional encoder network (AEN) to excavate rich semantic information from word embeddings.

Syntax-based methods: Recently, utilizing dependency information with GNNs has become an effective way for ABSA. Zhang et al. (2019) uses graph convolutional networks (GCN) to learn node representations from Dep.Tree. Tang et al. (2020) proposes a dependency graph enhanced dual-transformer network (DGEDT) by jointly considering representations from Transformers and corresponding dependency graph. Wang et al. (2020a) constructs aspect-oriented dependency trees and proposes R-GAT, extending the graph attention network to encode graphs with labeled edges. Li et al. (2021) proposes a dual graph convolutional networks (DualGCN) model, simultaneously considering syntax structures and semantic correlations. All above works use syntax information of Dep.Tree, which may introduce noise, as we said before. Thus, we exploit syntax information of Con.Tree with GNNs. Precisely, we follow the Con.Tree to aggregate information from words within the same phrases in a bottom-up way and capture *intra*-context information.

Moreover, some works resort to modeling aspect-aspect relations. Some (Hazarika et al., 2018; Majumder et al., 2018) adopt aspect representations to model relations by RNNs or memory networks, without utilizing context information. And some (Fan et al., 2018; Hu et al., 2019) propose alignment loss or orthogonal attention regulation to constrain aspect-level interactions, which fail when aspects have no explicit opinion expressions or multiple aspects share same opinion words. Recently, there are some works utilizing GNNs to model aspect relations. Liang et al. (2020) constructs an inter-aspect graph based on relative dependencies between aspects. Zhao et al. (2020) constructs a sentiment graph, where each node represents an aspect, and each edge represents the sentiment dependency relation. However, these works fail to explicitly use phrase segmentation information, such as conjunction words. Thus, we propose an aspect-context graph consisting of all aspects and phrase segmentation terms to model *inter*-context information.

GNNs with constituent tree: To our knowledge, we are the first work to utilize the constituent tree for ABSA task. But in aspect-category sentiment analysis task, which predicts sentiment polarity towards a given predefined category in the text, Li et al. (2020b) proposes a Sentence Constituent-Aware Network (SCAN) that generates representations of the nodes in Con.Tree. Unlike SCAN, we view parsed phrases as different spans of the input text instead of individual nodes. So we don't introduce any inner nodes of Con.Tree (*e.g.*, "NP","VP" of Figure 3) into the representation space, decreasing the computational overhead.

3 Methodology

3.1 Overview

Problem Statement. Let $\mathbf{s} = \{w_i\}_n$ and $\mathbf{A} = \{a_j\}_m$ be a sentence and a predefined aspect set, where n and m are the number of words in $\mathbf{s}$ and the number of aspects in $\mathbf{A}$, respectively. For each $\mathbf{s}$, $\mathbf{A_s} = \{a_i | a_i \in \mathbf{A}, a_i \in \mathbf{s}\}$ denotes the aspects contained in $\mathbf{s}$. We treat each multiple-word aspect as a single word for simplicity, so a_i also means the i-th word of $\mathbf{s}$. The goal of ABSA is to predict the sentiment polarity $y_i \in \{\text{positive, negative, neural}\}$ for each aspect $a_i \in \mathbf{A_s}$.

Architecture. As shown in Figure 4, our proposed architecture takes the sentence and all aspects that appear in the text as the input, and outputs the sentiment predictions of the aspects. It contains three components: 1) the ***intra*-context module** encodes the input $\{w_i\}$ to obtain aspect-specific representations of the target aspects, which contains two encoders: a context encoder that outputs contextual word representations and a syntax encoder that utilizes syntax information of the parsed constituent tree (or, and dependency tree). 2) the ***inter*-context module** includes a relation encoder applied to the constructed aspect-context graph to output relation-enhanced representations. The aspect-context graph composes all aspects of the given sentence and phrase segmentation terms obtained from a designed rule-based map function applied to the constituent tree. 3) the **sentiment classifier** takes output representations of the above two modules to make predictions.

3.2 *Intra*-Context Module

In this part, we utilize a context encoder and a syntax encoder to model the sentiment-aware context of every single aspect and generate aspect-specific representation for each aspect. Note that for multi-aspect sentences, we use this module multiple times, as each time deals with one aspect.

3.2.1 Context Encoder

We use BERT (Devlin et al., 2019) to generate contextual word representations. Given target aspect a_t, we follow **BERT-SPC** (Song et al., 2019) to construct a BERT-based sequence:

$$BERT\_seq_t = [\text{CLS}] + \{w_i\} + [\text{SEP}] + a_t + [\text{SEP}] , \quad (1)$$

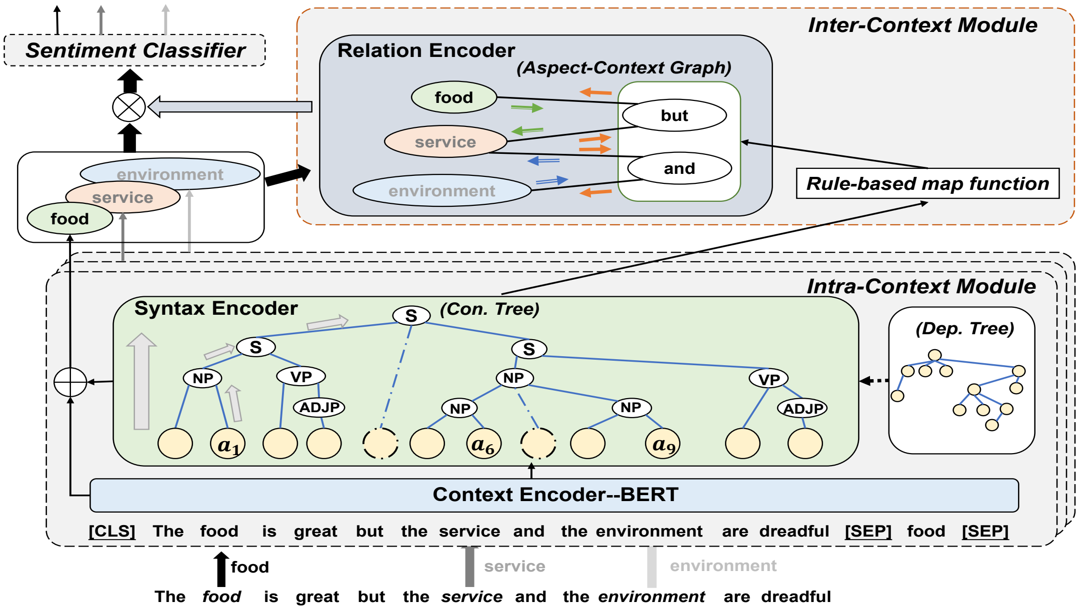

Figure 4: Overall architecture. It takes the sentence and all aspects as input and outputs sentiment predictions for all aspects. It has three components: 1) the ***intra*-context module** contains two encoders: a context encoder that outputs contextual word representations and a syntax encoder that utilizes syntax information of the parsed constituent tree (or, and dependency tree). Output representations from two encoders are fused to generate aspect-specific representations; 2) the ***inter*-context module** includes a relation encoder applied to the constructed aspect-context graph to obtain relation-enhanced representations. The aspect-context graph includes all aspects and phrase segmentation terms obtained from a designed rule-based map function applied to the constituent tree. 3) the **sentiment classifier** takes the outputs from two modules to make predictions.

Then, the output representation is obtained by,

$$h^t = \left\{h_0^t, h_1^t, \ldots, h_{n'}^t, \ldots, h_{n'+2+m'_t}^t\right\} \quad (2)$$

where n' and m' are lengths of input text and target aspect a_t after BERT tokenizer separately, h_0^t is "BERT pooling" vector representing the BERT sequence, h_i^t is the contextual representation of each token. Note that w_i may be split into multiple sub-words by BERT tokenizer. So we calculate the contextual representation of w_i as follows,

$$\hat{\mathbf{h}}_i^t = \frac{1}{|BertT(w_i)|} \sum_{k \in BertT(w_i)} h_k^t, \quad (3)$$

where $BertT(w_i)$ returns an index set of w_i's sub-words in BERT sequence, and $|\,|$ returns its length.

3.2.2 Syntax Encoder

The above representations only consider semantic information, so we propose a syntax encoder to utilize rich syntax information. Our syntax encoder is stacked by several designed Hierarchical Graph ATtention (HGAT) blocks, and each block consists of multiple graph attention (*i.e.,* GAT) layers that encode syntax information hierarchically under the guidance of the constituent tree (or, and the dependency tree). The key point is the construction of corresponding graphs.

Graph construction. As Figure 4 shows, we follow the syntax structure of Con.Tree in a bottom-up way. Each layer l of Con.Tree consists of several phrases $\{ph_u^l\}$ that compose the input text, and each phrase represents an individual semantic unit. *e.g.,* $\{ph^3\}$ in Figure 3 is {The food is great, but, the service and the environment are dreadful}. We construct corresponding graphs based on those phrases. *i.e.,* For layer l that consists of phrases $\{ph_u^l\}$, we construct the adjacent matrix $\mathbf{CA}$ that shows word connections:

$$\mathbf{CA}_{i,j}^l = \begin{cases} 1 & \text{if } w_i, w_j \text{ in same phrase of } \{ph_u^l\} \\ 0 & \text{otherwise} \end{cases}, \quad (4)$$

which is exemplified as *Con.Graphs* in Figure 5.

HGAT block. A HGAT block aims to encode syntax information into word representations hierarchically. As Figure 5 shows, a HGAT block is stacked by several GAT layers that utilize a masked self-attention mechanism to aggregate information from neighbors and a fully connected feed forward network to map representations to the same semantic space. Attention mechanism can handle the diversity of neighbors with higher weights assigned to more related words. It can be formulated as follows,

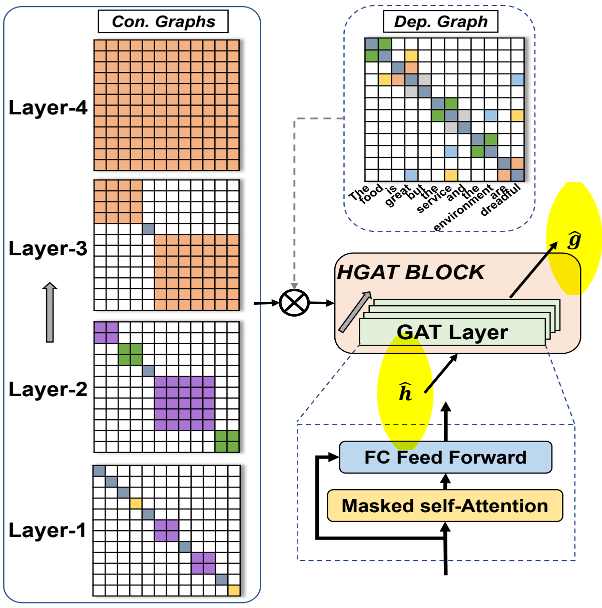

Figure 5: HGAT Block. It is stacked by several GAT layers, and each GAT layer is applied to the graph obtained from one layer of the constituent tree (or, and the dependency tree).

$$\hat{\mathbf{g}}_i^{t,l} = FC(\mathbf{g}_i^{t,l} + \hat{\mathbf{g}}_i^{t,l-1}), \quad (5)$$

$$\mathbf{g}_i^{t,l} = \|_{z=1}^{Z} \sigma \left(\sum_{j \in \mathcal{N}^{t,l}(i)} \alpha_{ij}^{lz} \mathbf{W}_g^{lz} \hat{\mathbf{g}}_j^{t,l-1} \right), \quad (6)$$

$$\alpha_{ij}^{lz} = \frac{\exp\left(f\left(\hat{\mathbf{g}}_i^{t,l-1}, \hat{\mathbf{g}}_j^{t,l-1}\right)\right)}{\sum_{j' \in \mathcal{N}^l(i)} \exp\left(f\left(\hat{\mathbf{g}}_i^{t,l-1}, \hat{\mathbf{g}}_{j'}^{t,l-1}\right)\right)}, \quad (7)$$

where $\mathcal{N}^l(i)$ is the set of neighbors of w_i in layer l, $\hat{\mathbf{g}}_i^{t,l}$ is the final representation of w_i in layer l, FC is fully connected feed forward network. $\mathbf{g}_i^{t,l}$ is the representation of w_i after masked self-attention mechanism. $||$ denotes vector concatenation. Z is the number of attention heads, σ is activation function. W_g^{lz} is trainable parameter of the zth head of layer l. f is a score function that measures the correlation of two words. Stacked HGAT block takes the output of previous one as the input, and the input of the first HGAT block is $\hat{\mathbf{h}}^t$. The output of syntax encoder is defined as $\hat{\mathbf{g}}^t$ for simplicity.

With dependency information. We also explore the fusion of two syntax information. Following previous works, we consider the Dep.Tree as an undirected graph and construct adjacent matrix **DA**, which is formulated as follows,

$$\mathbf{DA}_{i,j} = \begin{cases} 1 & \text{if } w_i, w_j \text{ link directly in Dep.Tree} \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

We consider three operations: **position-wise dot**, **position-wise add**, and **conditional position-wise add**. Each corresponding adjacent matrix **FA** is shown as follows,

A. position-wise dot. For each layer of Con.Tree, this operation only considers neighbors of the Dep.Tree that are also in the same phrase.

$$\mathbf{FA} = \mathbf{CA} \cdot \mathbf{DA} \quad (9)$$

B. position-wise add. For each layer of Con.Tree, this operation considers words in the same phrases and neighbors of the Dep.Tree. Some edges of Dep.Tree can shorten paths between aspect words and relevant opinion words, *e.g.*, "food" and "great" in Figure 3.

$$\mathbf{FA} = \mathbf{CA} + \mathbf{DA} \quad (10)$$

C. conditional position-wise add. This operation considers phrase-level syntax information of Con.Tree and clause-level syntax information of Dep.Tree. Specifically, it first deletes all dependency edges that are across clauses (*e.g.*, the edge between "great" and "dreadful" in Figure 2) and then conducts **position-wise add** operation with the remaining dependency edges.

$$\mathbf{FA} = \mathbf{CA} \oplus \mathbf{DA} \quad (11)$$

Thus, the output of the *intra*-context module contains both contextual information and syntax information, which is formulated as follows,

$$\mathbf{v}_t^{as} = \left[\hat{\mathbf{h}}_t^t + \hat{\mathbf{g}}_t^t ; h_0^t\right] \quad (12)$$

3.3 *Inter*-Context Module

The *intra*-context module ignores the mutual influence of aspects. Thus, in *inter*-context module, we construct an aspect-context graph to model the relations across aspects. This module only works for multi-aspect sentences, with aspect-specific representations of all aspects from *intra*-context module as input and outputs relation-enhanced representation of each aspect.

Phrase segmentation. Aspect relations can be revealed by some phrase segmentation terms, like conjunction words. Thus, we design a rule-based map function PS that returns phrase segmentation terms of two aspects: Given two aspects, it first finds their lowest common ancestor (LCA) in the Con.Tree, which contains information of two aspects and has the least irrelevant context. We call branches from LCA that between sub-trees which two aspects are separately in as "inner branches". PS returns all text words in the inner branches if they exist; else, it returns words between two aspects of the input text. It is formulated as follows,

$$PS(a_i, a_j) = \begin{cases} \{w_k\}, & \text{if } |Br(a_i, a_j)| = 0 \\ Br(a_i, a_j), & otherwise \end{cases}, \tag{13}$$

where $i < k < j$ and $Br(a_i, a_j)$ returns text words in the inner branches of a_i and a_j. *e.g.,* in Figure 3, given aspects food and service, the LCA node is S of Layer-4 that has three branches, with food in the first and service in the third. So "but" in the second branch (inner branch) is the phrase segmentation term that reflects sentiment relation of two aspects.

Aspect-context graph construction. We notice that the influence range of one aspect should be continuous, and the mutual influence of aspects attenuates with distance. Considering all aspect pairs introduces noise caused by long distance and increases computational overhead. So we only model relations across neighbor aspects. After extracting phrase segmentation terms of neighbor aspects by PS function, we construct an aspect-context graph by linking aspects with corresponding phrase segmentation terms to help infer relations. To distinguish the bi-directional relations over the aspect-context graph, we build two corresponding adjacent matrices. The first handles influence from aspects in odd-index among all aspects of the sentence, to neighbor even-index aspects, the second handles the opposite. An example is shown in Figure 6. Then, taking $\{v_t^{as}, t \in \mathbf{A_s}\}$ and corresponding phrase segmentation terms representations encoded by BERT as the input, the above HGAT blocks are applied as the relation encoder to obtain relation-enhanced representation v_t^{aa} for each aspect a_t.

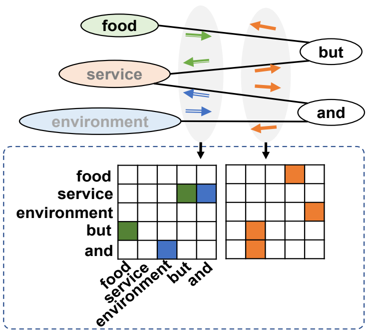

Figure 6: Example of an aspect-context graph and corresponding two adjacent matrices for distinguishing the bi-directional relations.

Dataset		Sentence-Level			Aspect-Level		
		Multi-Asp.	Single-Asp.	All	Pos.	Neg.	Neu.
Restaurant	Train	971	1009	1980	2164	807	637
	Test	315	284	599	727	196	196
Laptop	Train	538	916	1454	937	851	455
	Test	150	259	409	337	128	167
MAMS	Train	4297	0	4297	3380	2764	5042
	valid	500	0	500	403	325	604
	Test	500	0	500	400	329	607
Twitter	Train	0	6051	6051	1507	1528	3016
	Test	0	677	677	172	169	336

Table 1: Statistics of datasets. Multi-Asp., Single-Asp. indicate the number of sentences with multiple or single aspect; Pos., Neg., and Neu. show the number of aspects towards positive, negative and neutral label.

3.4 Training

The outputs of the ***intra*-context module** and ***inter*-context module** are combined to form the final representations, which are later fed to a fully connected layer (*i.e.,* sentiment classifier) with a softmax activation function, generating the probabilities over the three sentiment polarities:

$$\mathbf{o_t} = \mathbf{v}_t^{as} + \mathbf{v}_t^{aa}, \tag{14}$$

$$\mathbf{p(t)} = softmax(\mathbf{W_p o_t} + \mathbf{b_p}), \tag{15}$$

where $\mathbf{W_p}$, $\mathbf{b_p}$ are parameters of the classifier[1].

The loss is defined as the cross-entropy loss between golden polarity labels and predicted polarity distributions of all (sentence, aspect) pairs:

$$L(\theta)^{Sentiment} = -\sum_{s} \sum_{a_t \in A_s} loss(\mathbf{p(t)}, y(t)), \tag{16}$$

where a_t is the aspect and also the t-th word in s, $loss$ is the standard cross-entropy loss, θ represents model parameters.

4 Experiment

4.1 Datasets and Setup

We evaluate our models on four English dataset: Laptop, Restaurant datasets from SemEval2014 (Task 4) (Pontiki et al., 2014), MAMS (Jiang et al., 2019), and Twitter (Dong et al., 2014). Laptop and Restaurant contain both multi-aspect and single-aspect sentences. Each sentence in MAMS contains at least two aspects with different sentiments.

[1] In Eq14, $\mathbf{v}_t^{aa}$ is set to zero in single-aspect sentence.

Category	Model	Dataset							
		Restaurant		Laptop		MAMS		Twitter	
		Acc.(%)	F1.(%)	Acc.(%)	F1.(%)	Acc.(%)	F1.(%)	Acc.(%)	F1.(%)
w/o Syn.	BERT-SPC	84.46	76.98	78.99	75.03	82.82	81.90	73.55	72.14
	AEN-BERT	83.12	73.76	79.93	76.31	-	-	74.71	73.13
w/ Syn.	R-GAT	86.60	81.35	78.21	74.07	-	-	76.15	74.88
	RGAT+	86.68	80.92	80.94	78.20	84.52	83.74	76.28	75.25
	DGEDT	86.30	80.00	79.80	75.60	-	-	77.90	75.40
	DualGCN	87.13	81.16	81.80	78.10	-	-	77.40	76.02
	SDGCN	83.57	76.47	81.35	78.34	-	-	-	-
	InterGCN	87.12	81.02	82.87	79.32	-	-	-	-
Ours	BiSyn-GAT	87.49	81.63	82.44	79.15	84.90	84.43	**77.99**	**76.80**
	BiSyn-GAT+	**87.94**	**82.43**	**82.91**	**79.38**	**85.85**	**85.49**		

Table 2: Performance comparison of models on four datasets. The best are in **bold**, and second-best are underlined.

Category	Ablation	Dataset							
		Restaurant		Laptop		MAMS		Twitter	
		Acc.(%)	F1.(%)	Acc.(%)	F1.(%)	Acc.(%)	F1.(%)	Acc.(%)	F1.(%)
w/o AA	w/o syn. & dep.(BERT+)	84.99	78.51	79.11	75.76	82.71	82.22	75.48	74.54
	w/o con.	86.42	80.10	80.22	76.42	83.38	82.90	76.51	75.29
	w/o dep.	86.60	81.51	81.80	78.48	84.58	84.09	76.81	75.86
	con.$\times$dep.	86.86	80.82	80.85	77.27	84.21	83.76	76.51	75.37
	con.+dep.	86.86	81.59	82.12	78.93	84.73	84.14	77.40	76.39
	con.$\oplus$dep. (BiSyn-GAT)	87.49	81.63	82.44	79.15	84.90	84.43	**77.99**	**76.80**
w/ AA	con.+dep.	87.76	82.18	82.75	79.16	85.48	85.05	-	-
	con.$\oplus$dep. (BiSyn-GAT+)	**87.94**	**82.43**	**82.91**	**79.38**	**85.85**	**85.49**	-	-

Table 3: Ablation study. Notations "con." and "dep." represent syntax information from constituent tree and dependency tree, respectively. $\times, +, \oplus$ represent the position-wise dot, position-wise add, conditional position-wise add operations, respectively, when fusing two syntax information. "AA" represents modeling aspect-aspect relations. The best performances are in **bold**, and second-best are underlined.

Twitter contains only one-aspect sentences. Dataset statistics are shown in Table 1.

We adopt SuPar[2] as parser. Specifically, we use CRF constituency parser (Zhang et al., 2020) to get the constituent tree; and following previous works (Wang et al., 2020a; Bai et al., 2020), we use deep Biaffine Parser (Dozat and Manning, 2017) to get the dependency tree. Our context encoder is BERT-base-uncased [3] model. Adam optimizer is adopted with a learning rate 2×10^{-5} and a L_2 regulation 10^{-5} for model training. Number of GAT layers of one HGAT block is 3, and number of HGAT blocks is in range [1,3] on different datasets. "Accuracy" and "Macro-Averaged F1" are evaluation metrics. More details are in Appendix A.

4.2 Baselines

We compare our model with the following models:

1) Syntax-free baselines: **BERT-SPC** (Song et al., 2019), **AEN-BERT** (Song et al., 2019);

2) Syntax-based baselines: **R-GAT** (Wang et al., 2020a), **RGAT+** (Bai et al., 2020), **DGEDT** (Tang et al., 2020), **DualGCN** (Li et al., 2021);

3) Baselines that model aspect-aspect relations: **SDGCN-BERT** (Zhao et al., 2020), **InterGCN** (Liang et al., 2020);

[2]https://github.com/yzhangcs/parser

[3]https://github.com/huggingface/transformers

Model		Dataset			
		Restaurant		MAMS	
		Acc.(%)	F1.(%)	Acc.(%)	F1.(%)
	BiSyn-GAT	87.49	81.63	84.90	84.43
aspect-context graph	w/ Bi-relation	**87.94**	**82.43**	**85.85**	**85.49**
	w/o Bi-relation	87.85	82.27	85.10	84.69
aspect graph	adjacent	87.49	81.69	85.10	84.61
	Bi-adjacent	87.40	81.53	85.18	84.74
	global	87.49	81.70	85.32	84.88

Table 4: Performance comparison of aspect-context graph variants on Restaurant and MAMS dataset. The best performances are in **bold**.

Ours are also syntax-based, including:

a) **BiSyn-GAT+**: our full model, which contains the *intra*-context module that combines two syntax information by **conditional position-wise add** operation, *inter*-context module, and sentiment classifier to make predictions;

b) **BiSyn-GAT**: full model without *inter*-context module;

Baselines and our models are all BERT-based.

4.3 Main Results

Table 2 shows results of the baselines and our models. For fairness of comparison, we present the reported results of those baselines. Observations are: 1) Our proposed models outperform most baselines, and our full model ***BiSyn-GAT+*** achieves state-

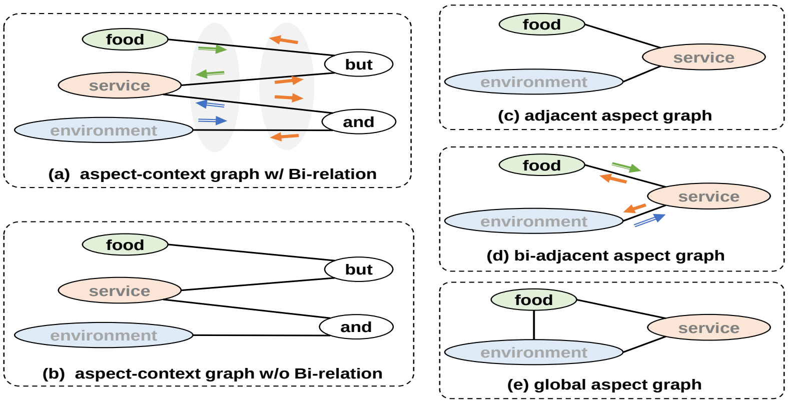

Figure 7: Illustrations of variants when investigating the effects of aspect-context graph.

Model	Parser	Restaurant		MAMS	
		Acc.(%)	F1.(%)	Acc.(%)	F1.(%)
Base		84.99	78.51	82.71	82.22
w/o dep.	Stanford Parser	86.51	81.34	84.51	84.06
	SuPar	86.60	81.51	84.58	84.09
BiSyn-GAT	Stanford Parser	86.66	81.56	84.88	84.31
	SuPar	87.49	81.63	84.90	84.43
BiSyn-GAT+	Stanford Parser	87.84	82.39	85.78	85.40
	SuPar	87.94	82.43	85.85	85.49

Table 5: Experiments results with different parsers. w/o dep. is one variant of BiSyn-GAT, only using constituent information.

of-the-art performances in all datasets, especially 1.27 and 1.75 F1 improvements on Restaurant and MAMS. 2) Models with syntax information outperform those without, which means syntax structure is helpful. 3) Our models show superiority to those that only use dependency information, which implies that constituent tree can provide profitable information. 4) ***BiSyn-GAT+*** shows consistent improvement compared to ***BiSyn-GAT***, which means modeling aspect-aspect relations can improve performance, especially when more multi-aspect sentences are available, *e.g.,* 0.8 and 1.06 F1 improvements on Restaurant and MAMS.

4.4 Ablation Study

We also conduct an ablation study to verify the effectiveness of our proposed method. The results are shown in Table 3. We set the context encoder of our model as the base model, *i.e., BERT+*. The observations are that: 1) *BERT+* achieves the lowest performance, which shows syntax information is helpful in ABSA task. 2) In category *w/o AA*, **w/o con.** is inferior to **w/o dep.**, which means syntax information of Con.Tree is useful. Moreover, the comparison between **w/o con.** and **con.**$\times$**dep.** verifies that some dependency edges that cross the phrases indeed bring noise, as the former considers all dependency edges and the latter ignores those across phrases obtained from Con.Tree for each layer. 3) Fusing two syntax information in the proper ways can boost performance. In category *w/o AA*, **con.+dep.** and **con.**$\oplus$**dep.** both outperform **w/o dep.** and **w/o con.** in all datasets. However, **con.**$\times$**dep.** is inferior to **w/o dep.**. One possible reason is that the position-wise dot operation ignores most connections within phrases, causing the graphs to be more sparse. It also verifies that words within the same phrases of Con.Tree are essential for aligning aspects and corresponding opinions. 4) Modeling aspect-aspect relations is beneficial from the comparison between *w/ AA* and *w/o AA*, especially in Restaurant and MAMS that contain more multi-aspect sentences.

5 Effects of Aspect-context graph

We also investigate the effects of our bi-relational modeling of the proposed aspect-context graph. Firstly, we use **BiSyn-GAT** as base model to see whether the approach modeling aspects relations improves the performance; Secondly, based on our proposed aspect-context graph, we consider two variants: (a) **w/ Bi-relation**, a directed one that distinguishes the influence one aspect imposes on other aspects and is received from other aspects, *i.e.,* our full model BiSyn-GAT+; (b) **w/o Bi-relation**, an undirected one that ignores the direction of the influence; Thirdly, inspired by Zhao et al. (2020), we define the aspect graph as the graph with all aspects as its nodes, *i.e.,* our aspect-context graph without any segmentation terms. Based on the aspect graph, we propose three variants: (c) **adjacent aspect graph**, an undirected one where neighbor aspects are connected; (d) **bi-adjacent aspect graph**, a directed one where neighbor aspects are connected; (e) **global aspect graph**, an undirected one where all aspects are connected; The above five variants are illustrated in Figure 7. Experimental

Sentences	Aspects	BiSyn-GAT	BiSyn-GAT+
it doesn't look like much on the **outside**$_{\mathbf{neg}}$, but the minute you walk inside, it's a whole other **atmosphere**$_{\mathbf{pos}}$.	outside atmosphere	neu ✗ pos ✓	neg ✓ pos ✓
while the **service**$_{\mathbf{neg}}$ and **setting**$_{\mathbf{neg}}$ were average , the **food**$_{\mathbf{pos}}$ was excellent.	service setting food	neg ✓ neu ✗ pos ✓	neg ✓ neg ✓ pos ✓
food was average, the **appetizers**$_{\mathbf{pos}}$ were better than the **main courses**$_{\mathbf{neu}}$.	appetizers main courses	pos ✓ pos ✗	pos ✓ neu ✓
i have no complaints about the **wait**$_{\mathbf{pos}}$ or the **service**$_{\mathbf{pos}}$ but the **pizza**$_{\mathbf{neg}}$ was bit at all something to write home about.	wait service pizza	neu ✗ neg ✗ neg ✓	pos ✓ pos ✓ neg ✓

Table 6: Predictions from *BiSyn-GAT* and *BiSyn-GAT+*. The notations pos, neg, and neu in the table represent positive, negative, and neutral. For each sentence, the aspects are displayed in bold, with golden sentiment polarities as the subscripts. The phrase segmentation words are shown underline between the corresponding two aspects. False predictions are marked with ✗while true predictions are marked with ✓.

results are shown in Table 4 and we can observe that: 1) w/ Bi-relation (*i.e.,* BiSyn-GAT+) outperforms w/o Bi-relation consistently, which indicates distinguishing the bi-relational influences is beneficial; 2) Overall, aspect-context graph shows superiority compared with aspect graph, which means the phrase segmentation terms can help model aspects relations; 3) Unlike in aspect-context graph, bi-adjacent aspect graph does not guarantee performance improvement compared with adjacent aspect graph, which reflects the importance of phrase segmentation terms when modeling aspect-aspect relations; 4) Overall, global aspect graph performs better than adjacent aspect graph, which is correlated with the results in Zhao et al. (2020); 5) In Restaurant dataset, adjacent aspect graph and global aspect graph show comparable performance. One possible reason is that the number of samples that contain at least three aspects is very limited, as shown in Table 8 of Appendix. And adjacent aspect graph equals global aspect graph when faced with two aspects.

5.1 Effects of Parsing

We conduct experiments to study the influence of paring accuracy on model performance. Two parsers are selected: (a) Stanford Parser (Manning et al., 2014), a well-known toolkit; it has transition-based dependency parser (Chen and Manning, 2014) and shift-reduce constituency parser (Zhu et al., 2013); (b) SuPar, which RGAT+ (Bai et al., 2020) and our proposed models adopt; it has deep biaffine dependency parser (Dozat and Manning, 2017) and neural CRF constituency parser (Zhang et al., 2020). Generally, SuPar has better parsing performances than Stanford Parser. We use BERT+ as the base model and compare the performance of model **w/o dep**, **Bisyn-GAT**, **BiSyn-GAT+** when using different parsers. The results are shown in Table 5. Observations are that: 1) With Stanford Parser, our models can also achieve good performance. 2) Models with SuPar perform better than models with Stanford Parser, which is correlated with the parsing accuracy of two parsers.

5.2 Case Study

As shown in Figure 6, we present four examples to help better understand our proposed model, especially *inter*-context module when faced with complex sentences. The first is a comparative sentence with two clauses connected by the conjunction "but". Both models make correct predictions for **atmosphere**. However, *BiSyn-GAT* predicts wrong over **outside** while *BiSyn-GAT+* still makes a correct prediction, which show the *inter*-context module correctly captures the reversed sentiment relation between **outside** and **atmosphere** by phrase segmentation terms ", but". The rest examples all show that *inter*-context module can use relations across aspects to help correct the predictions.

6 Conclusion

In this paper, we propose the BiSyn-GAT+ framework to model the sentiment-aware context of each aspect and sentiment relations across aspects for learning by fully exploiting the syntax information of the constituent tree. It includes two well-designed modules: 1) *intra*-context module that fuses related semantic and syntax information hierarchically; 2) *inter*-context module that models relations across aspects with the constructed aspect-context graph. To the best of our knowledge, it is the first work to exploit the constituent tree with GNNs for the ABSA task. Moreover, our proposed model achieves state-of-the-art performances on four benchmark datasets.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant No.61602197, Grant No.L1924068, Grant No.61772076, in part by CCF-AFSG Research Fund under Grant No.RF20210005, and in part by the fund of Joint Laboratory of HUST and Pingan Property & Casualty Research (HPL). The authors would also like to thank the anonymous reviewers for their comments on improving the quality of this paper.

References

Xuefeng Bai, Pengbo Liu, and Yue Zhang. 2020. Investigating typed syntactic dependencies for targeted sentiment classification using graph attention neural network. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:503–514.

Chenhua Chen, Zhiyang Teng, and Yue Zhang. 2020. Inducing target-specific latent structures for aspect sentiment classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5596–5607, Online. Association for Computational Linguistics.

Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750, Doha, Qatar. Association for Computational Linguistics.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Feifan Fan, Yansong Feng, and Dongyan Zhao. 2018. Multi-grained attention network for aspect-level sentiment classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3433–3442, Brussels, Belgium. Association for Computational Linguistics.

Devamanyu Hazarika, Soujanya Poria, Prateek Vij, Gangeshwar Krishnamurthy, Erik Cambria, and Roger Zimmermann. 2018. Modeling inter-aspect dependencies for aspect-based sentiment analysis. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 266–270, New Orleans, Louisiana. Association for Computational Linguistics.

Mengting Hu, Shiwan Zhao, Li Zhang, Keke Cai, Zhong Su, Renhong Cheng, and Xiaowei Shen. 2019. CAN: Constrained attention networks for multi-aspect sentiment analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4601–4610, Hong Kong, China. Association for Computational Linguistics.

Qingnan Jiang, Lei Chen, Ruifeng Xu, Xiang Ao, and Min Yang. 2019. 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)*, pages 6280–6285, Hong Kong, China. Association for Computational Linguistics.

Tian Lan, Xian-Ling Mao, Wei Wei, Xiaoyan Gao, and Heyan Huang. 2020. Pone: A novel automatic evaluation metric for open-domain generative dialogue systems. *ACM Trans. Inf. Syst.*, 39(1).

Qintong Li, Hongshen Chen, Zhaochun Ren, Pengjie Ren, Zhaopeng Tu, and Zhumin Chen. 2020a. EmpDG: Multi-resolution interactive empathetic dialogue generation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4454–4466, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Ruifan Li, Hao Chen, Fangxiang Feng, Zhanyu Ma, Xiaojie Wang, and Eduard Hovy. 2021. 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)*, pages 6319–6329, Online. Association for Computational Linguistics.

Yuncong Li, Cunxiang Yin, and Sheng hua Zhong. 2020b. Sentence constituent-aware aspect-category sentiment analysis with graph attention networks. In *NLPCC*.

Zheng Li, Ying Wei, Yu Zhang, Xiang Zhang, and Xin Li. 2019. Exploiting coarse-to-fine task transfer for aspect-level sentiment classification. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 4253–4260. AAAI Press.

Bin Liang, Rongdi Yin, Lin Gui, Jiachen Du, and Ruifeng Xu. 2020. 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*, pages 150–161, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Navonil Majumder, Soujanya Poria, Alexander Gelbukh, Md. Shad Akhtar, Erik Cambria, and Asif Ekbal. 2018. IARM: Inter-aspect relation modeling with memory networks in aspect-based sentiment analysis. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3402–3411, Brussels, Belgium. Association for Computational Linguistics.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, Dublin, Ireland. Association for Computational Linguistics.

Minghui Qiu, Xinjing Huang, Cen-Chieh Chen, Feng Ji, Chen Qu, Wei Wei, Jun Huang, and Yin Zhang. 2021. Reinforced history backtracking for conversational question answering. In *AAAI*.

Youwei Song, Jiahai Wang, Tao Jiang, Zhiyue Liu, and Yanghui Rao. 2019. Attentional encoder network for targeted sentiment classification. *ArXiv*, abs/1902.09314.

Hao Tang, Donghong Ji, Chenliang Li, and Qiji Zhou. 2020. Dependency graph enhanced dual-transformer structure for aspect-based sentiment classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6578–6588, Online. Association for Computational Linguistics.

Kai Wang, Weizhou Shen, Yunyi Yang, Xiaojun Quan, and Rui Wang. 2020a. Relational graph attention network for aspect-based sentiment analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3229–3238, Online. Association for Computational Linguistics.

Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based LSTM for aspect-level sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 606–615, Austin, Texas. Association for Computational Linguistics.

Ziyang Wang, Wei Wei, Gao Cong, Xiao-Li Li, Xian-ling Mao, and Minghui Qiu. 2020b. Global context enhanced graph neural networks for session-based recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 169–178. ACM.

Wei Wei, Gao Cong, Xiaoli Li, See-Kiong Ng, and Guohui Li. 2011. Integrating community question and answer archives. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, USA, August 7-11, 2011*. AAAI Press.

Wei Wei, Jiayi Liu, Xianling Mao, Guibing Guo, Feida Zhu, Pan Zhou, and Yuchong Hu. 2019. Emotion-aware chat machine: Automatic emotional response generation for human-like emotional interaction. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, pages 1401–1410. ACM.

Wei Wei, Jiayi Liu, Xianling Mao, Guibing Guo, Feida Zhu, Pan Zhou, Yuchong Hu, and Shanshan Feng. 2021. Target-guided emotion-aware chat machine. *ACM Transactions on Information Systems (TOIS)*, 39(4):1–24.

Jun Yang, Runqi Yang, Chongjun Wang, and Junyuan Xie. 2018. Multi-entity aspect-based sentiment analysis with context, entity and aspect memory. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 6029–6036. AAAI Press.

Kaicheng Yang, Hua Xu, and Kai Gao. 2020. CM-BERT: cross-modal BERT for text-audio sentiment analysis. In *MM '20: The 28th ACM International Conference on Multimedia, Virtual Event / Seattle, WA, USA, October 12-16, 2020*, pages 521–528.

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

Lei Zhang, Shuai Wang, and Bing Liu. 2018. Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):e1253.

Yu Zhang, Houquan Zhou, and Zhenghua Li. 2020. Fast and accurate neural CRF constituency parsing. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 4046–4053. ijcai.org.

Pinlong Zhao, Linlin Hou, and Ou Wu. 2020. Modeling sentiment dependencies with graph convolutional networks for aspect-level sentiment classification. *Knowledge-Based Systems*, 193:105443.

Sen Zhao, Wei Wei, Ding Zou, and Xianling Mao. 2022. Multi-view intent disentangle graph networks for bundle recommendation. *ArXiv preprint*, abs/2202.11425.

Muhua Zhu, Yue Zhang, Wenliang Chen, Min Zhang, and Jingbo Zhu. 2013. Fast and accurate shift-reduce constituent parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 434–443, Sofia, Bulgaria. Association for Computational Linguistics.

Con. Tree Depth	Dataset								
	Restaurant		Laptop		MAMS			Twitter	
	Train	Test	Train	Test	Train	Valid	Test	Train	Test
1	177	68	206	84	208	16	19	1215	117
2	369	135	724	247	1301	152	141	1066	**147**
3	**462**	**148**	**936**	**312**	**2265**	244	261	**1186**	123
4	363	108	612	202	2085	**276**	**292**	947	96
5	311	75	429	116	1761	203	194	677	79
6	237	40	266	73	1211	141	157	414	57
7	136	27	205	41	901	99	117	246	23
8	108	10	106	18	545	81	65	145	22
9	59	8	43	14	380	57	34	86	8
≥ 10	60	13	81	12	529	63	56	69	5
MAX.	18	13	17	13	19	17	15	14	11

Table 7: Depth distribution of parsed constituent trees on four datasets. The maximums are in **bold**. The last row lists the max tree depth of each dataset.

Multi. Aspect Distribution	Dataset						
	Restaurant		Laptop		MAMS		
	Train	Test	Train	Test	Train	Valid	Test
2	555	192	343	101	2568	285	264
3	261	73	137	33	1169	136	173
4	103	31	40	9	364	55	45
5	32	14	9	6	126	16	10
6	11	3	5	1	48	5	5
7	5	1	3	-	13	2	-
8	3	-	-	-	6	-	1
9	1	-	-	-	1	-	-
10	-	-	-	-	1	1	1
11	-	-	-	-	1	1	1
13	-	1	1	-	-	-	

Table 8: Multi.aspect distribution of three datasets.

A Dataset and Implementation Detail

A.1 Statistics of constituent tree depth

Table 7 shows more detailed statistics about four benchmark datasets at the aspect level. We define the "constituent tree depth" as the number of nodes in the path from the aspect term node to the root node in the Con.Tree. It means we treat the layer that the aspect term is in as the bottom layer for constituent graph construction and drop layers below it. The aspect term has no other neighbors in those layers and thus fails to update its representation through the graph encoder. According to the constituent tree depth statistics, we set the number of GAT layers of one HGAT block in the syntax encoder to 3, the most common depth.

A.2 Multi-aspect Distribution of datasets

Table 8 shows the multi-aspect distribution of the Restaurant, Laptop, and MAMS datasets. This can explain the improvement of BiSyn-GAT+ compared to BiSyn-GAT on different datasets: MAMS > Restaurant > Laptop. MAMS contains the most multi-aspect sentences that our proposed *Inter-context* module can fully utilize.

A.3 Training Detail

The numbers of parameters of BiSyn-GAT and BiSyn-GAT+ are 112M and 233M. Each epoch takes about 60s or 70s in RTX 2080 Ti. We test the model that performs best on validation data, and for datasets without official validation data, we follow the dataset settings of previous work (Bai et al., 2020). We use the grid search to find the best parameters for our model and report the maximum results. The number of HGAT blocks within our relation encoder is in range [1,3] on different datasets and the number of its inner GAT layers is set to 2; the dropout rate is 0.1 for the input and output and is in the range [0.2, 0.7] between layers; In each HGAT block of our syntax encoder, for samples with fewer constituent tree layers, we only adopt the same number of GAT layers to encode; for samples with more constituent tree layers, we prune them to three layers.

B Discussion about phrase segmentation term

We firstly provide more cases about the phrase segmentation terms in this section. For each case, the aspects are displayed in **bold** and phrase segmentation words are underlined between the corresponding two aspects:

1) However, we went for **lunch** and were the only ones eatting there and yet the **service** seemed eager for use to be done and to get out.

2) We were so excited since I was reading great review of this **place**, however we were disappointed with the **taste** of the **food**.

3) Then the **manager** gave us **lemon juice** instead of **ceasar dressing** for a ceasar salad which ruined the salad.

4) The only drawback was slow **service**, but the **food** and **ambiance** are so nice that your **wait** is a) pleasant and b) worth it.

5) Compared to the **soup** of average taste, the **rice** is better in this restaurant.

The top 4 cases show that our approach can capture words, such as "and", "but", "yet", "however", "instead of" to help infer aspects relations.

However, we also notice there is a limitation of our method: it can only find the phrase segmentation terms within the two aspects, failing to capture some important words indicative of relations that appear in other locations. *e.g.,* in case 5), our approach capture "," instead of "compared to", while only the latter can show the reversed sentiment of two aspects. We leave this problem as the future work, considering that our current approach is simple and can also achieve good performance.

C Limitations and future work

This section discusses some improvements that can be made in future work. 1) Our full model adopts two BERT encoders, one in *Intra*-context module for encoding input text and aspects and one in *Inter*-context module for encoding the phrase segmentation terms. The pros are that our *Inter*-context can easily generalize to other ABSA models, taking their output aspect representations and generating the relation enhanced representations. However, this causes the parameters of BiSyn-GAT+ up to 233M. We will consider other encoding strategies instead of simply using another BERT; 2) We notice that the label information from Con.Tree can also provide valuable information, *e.g.,* NP node and VP node, which together form the S node, may contain the aspect term and corresponding opinion words separately, as shown in Figure 3. It is worth trying to utilize more information from Con.Tree, and we will continue to explore it in future work.