Learn from Syntax: Improving Pair-wise Aspect and Opinion Terms Extraction with Rich Syntactic Knowledge

Shengqiong Wu[1,*], **Hao Fei**[1,*], **Yafeng Ren**[2], **Donghong Ji**[1,†] and **Jingye Li**[1]
[1]Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan, China
[2]Guangdong University of Foreign Studies, Guangzhou, China
{whuwsq, hao.fei, renyafeng, dhji, theodorelee}@whu.edu.com

Abstract

In this paper, we propose to enhance the pair-wise aspect and opinion terms extraction (PAOTE) task by incorporating rich syntactic knowledge. We first build a syntax fusion encoder for encoding syntactic features, including a label-aware graph convolutional network (LAGCN) for modeling the dependency edges and labels, as well as the POS tags unifiedly, and a local-attention module encoding POS tags for better term boundary detection. During pairing, we then adopt Biaffine and Triaffine scoring for high-order aspect-opinion term pairing, in the meantime re-harnessing the syntax-enriched representations in LAGCN for syntactic-aware scoring. Experimental results on four benchmark datasets demonstrate that our model outperforms current state-of-the-art baselines, meanwhile yielding explainable predictions with syntactic knowledge.

1 Introduction

Fine-grained aspect-based sentiment analysis (ABSA), which aims to analyze people's detailed insights towards a product or service, has become a hot research topic in natural language processing (NLP). The extraction of aspect terms (AT) extraction and opinion terms (OT) as two fundamental subtasks of ABSA have emerged [Wang *et al.*, 2017; Xu *et al.*, 2018; Fan *et al.*, 2019; Chen and Qian, 2020]. In later research, the aspect and opinion terms co-extraction has received much attention for the exploration of mutual benefits in between [Wang *et al.*, 2017; Dai and Song, 2019]. However, these extraction methods do not consider AT and OT as pairs. More recently, some efforts are devoted to detecting the pair of the correlated aspect and opinion terms jointly, namely pair-wise aspect and opinion terms extraction (PAOTE) task [Zhao *et al.*, 2020; Wu *et al.*, 2020a; Chen *et al.*, 2020], as illustrated in Figure 1. Existing works perform end-to-end PAOTE based on joint learning methods for better task performances [Zhao *et al.*, 2020; Wu *et al.*, 2020a; Chen *et al.*, 2020]. Unfortunately, there

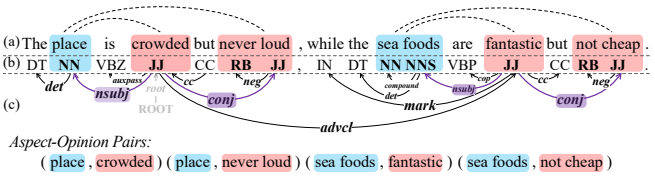

Figure 1: Illustration of pair-wise aspect and opinion terms extraction based on an example sentence (a) with the corresponding part-of-speech tags (b) and syntactic dependency structures (c).

are still some characteristics of PAOTE fallen out of the consideration of prior works.

Firstly, the linguistic part-of-speech (POS) tag features are an overlooked potential performance enhancer. Intuitively, POS tags entail the boundary information between neighbors of spans, which can essentially promote the recognition of aspect and opinion terms. Secondly, the syntactic structure knowledge is highly crucial to PAOTE, i.e., helping to capture some long-range syntactic relations that are obscure from the surface form alone. Yet only the syntactic dependency edge features are utilized in prior works for ABSA (i.e., the tree structure), without considering the syntactic dependency label features [Zhang *et al.*, 2019]. We note that the syntactic labels also provide key clues for supporting the underlying reasoning. Intuitively, the dependency arcs with different labels carry distinct evidence in different degrees. As exemplified in Figure 1, Compared with other arcs within the dependency structure, the ones with '*nsubj*' and '*conj*' can bring the most characteristic clues for facilitating the inference of the aspect-opinion pairs.

Another observation is that the considerable numbers of overlapping[1] aspect-opinion pairs (around 24.42% in our data) may largely influence the task performances. Essentially, those aspect-opinion pairs within one overlapping structure may share some mutual information. Notwithstanding, the first-order scoring paradigm has been largely employed in the current graph-based PAOTE models [Zhao *et al.*, 2020; Wu *et al.*, 2020a; Chen *et al.*, 2020], considering only one single potential aspect-opinion pair at a time when making scoring. This inevitably results in local short-term feature combination and leaves the underlying common structural

*Equal contribution
†Corresponding Author

[1]An aspect or opinion term in one pair is simultaneously involved in other pair(s), as in Figure 1.

interactions unused. Hence, how to effectively model the overlapping structure during the term pairing remains unexplored.

In this paper, we aim to address all the aforementioned challenges by presenting a novel joint framework for PAOTE. Figure 2 shows the overall framework. First, we propose a syntax fusion encoder (namely SynFue) for encoding syntactic features (cf. Figure 3), where a label-aware graph convolutional network (LAGCN) models dependency edges and labels as well as POS tags, and the local-attention module encodes POS tags. By capturing rich syntactic knowledge in this manner, SynFue is able to produce span terms more accurately, and on the other hand, it encourages sufficient interactions between syntactic structures and term pair structures. During pairing, we then perform high-order scoring for each candidate aspect-opinion term pair via a Triaffine scorer [Carreras, 2007], which can model the triadic relations of the overlapping term structures with a broader viewpoint. To enhance the semantic pairing, we further consider a syntactic-aware scoring, re-harnessing the syntax-enriched representations in LAGCN. Finally, our system outputs all valid aspect-opinion term pairs based on the overall potential scores.

To sum up, our contributions are three-fold.

★ We for the first time in literature propose to incorporate rich syntactic and linguistic knowledge for improving the PAOTE task. We propose a LAGCN to encode the dependency trees with labels as well as POS tags in a unified manner. Also, we promote the term boundary recognition by modeling the POS features via a local attention mechanism.

★ We present a high-order joint solution for aspect-opinion pairing with a Triaffine scorer, fully exploring the underlying mutual interactions within the overlapping pair structures. The intermediate syntax-enriched representations yielded from LAGCN are re-exploited for further syntactic-aware scoring.

★ Our method attains state-of-the-art performances on four benchmark datasets for PAOTE. Further analysis reveals that our method can effectively leverage rich syntactic information, and capture the correlations between syntactic structures and aspect-opinion pair structures.

2 Model

As illustrated in Figure 2, our system is built based on the current best-performing span graph-based model [Zhao *et al.*, 2020; Eberts and Ulges, 2020]. The model first takes as inputs the contextualized word representation from the BERT language model [Devlin *et al.*, 2019]. Next, syntactic dependencies and POS tags are injected into the syntax fusion encoder. We then perform term type classification and filtering based on the term representations from the token representations. In the pairing stage, we measure the term-term pairs with the potential scores including high-order scores and syntactic-aware scores, based on which the final pairs will be output. Given an input sentence s=$\{w_1, \cdots, w_T\}$, our system is expected to produce a set of aspect-opinion pairs P=$\{p_1(a,o), \cdots, p_k(a,o)\} \subset A \times O$. A=$\{a_1, \cdots, a_N\}$ is all possible aspect terms, where a_n can be a single word or a phrase, denoted as a_n=$\{w_i, \cdots, w_j\}$. Likewise, O=$\{o_1, \cdots, o_M\}$ is all the possible opinion terms, where o_m=$\{w_i, \cdots, w_j\}$ is a term span.

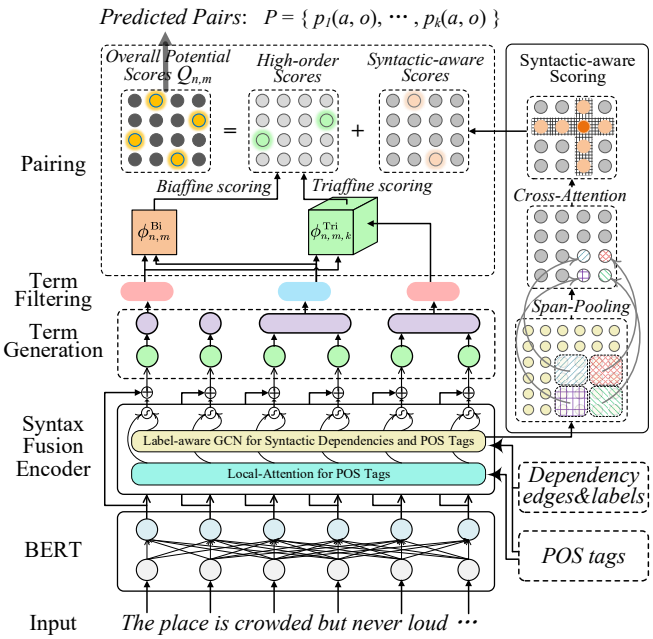

Figure 2: Overview of our proposed framework.

2.1 Word Representation from BERT

The BERT language model has been proven superior in building contextualized representations for various NLP tasks [Ding *et al.*, 2020; Eberts and Ulges, 2020; Zhao *et al.*, 2020]. Hence we utilize BERT as the underlying encoder to yield the basic contextualized word representations:

$$\{\boldsymbol{v}_1, \cdots, \boldsymbol{v}_T\} = \text{BERT}(\{w_1, \cdots, w_T\}), \quad (1)$$

where v_t is the output representation for the word w_t.

2.2 Syntax Fusion Encoder

We inject three types of external syntactic sources, i.e., dependency edges and labels, and POS tags, into our syntax fusion encoder (SynFue). SynFue (with total L layers) consists of a local attention for POS tags and a label-aware GCN for all syntactic inputs at each layer (cf. Figure 3).

Local-attention for encoding POS tags POS tags, as the major word-level linguistic features, provide potential clues for boundary recognition of term spans [Nie *et al.*, 2020]. Instead of adopting the vanilla hard attention that encodes the whole sequence-level information, we encode POS tags via a local attention mechanism [Luong *et al.*, 2015], which is more capable of capturing the local contexts for phrasal term spans. Technically, for each word w_t in s, we mark its corresponding POS tag as w_t^p, and obtain its POS embedding $\boldsymbol{x}_t^p$. At the l-th layer, the local attention operation is performed at a scope of d window size:

$$\boldsymbol{e}_t^{p,l} = \textstyle\sum_{i=t-d}^{t+d} \gamma_{t,i}^l \, \boldsymbol{x}_i^p \,, \quad (2)$$

$$\gamma_{t,i}^l = \frac{\exp\left(\boldsymbol{W}_1[\boldsymbol{e}_i^{l-1}; \boldsymbol{x}_i^p]\right)}{\sum_{j=t-d}^{t+d} \exp\left(\boldsymbol{W}_1[\boldsymbol{e}_j^{l-1}; \boldsymbol{x}_j^p]\right)} \,, \quad (3)$$

where $[;]$ denotes the concatenation operation, $\boldsymbol{W}_1$ is the learnable parameters and $\boldsymbol{e}_t^{p,l}$ is the output representations.

Label-aware GCN for rich syntactic features Previous studies employ GCN [Marcheggiani and Titov, 2017] to encode purely the dependency structural edges, while they fail to model the syntactic dependency labels leeched on to the edges, but also ignore the POS category information. We note that these syntactic features should be navigated simultaneously in a unified manner, as they together essentially describe the complete syntactic attributes in different perspectives. We here propose a label-aware GCN (LAGCN) to accomplish it. Given the input sentence s with its corresponding dependency edges and labels, and POS tag embeddings $\boldsymbol{x}_t^p$, we define an adjacency matrix $\{b_{t,j}\}_{T\times T}$ for dependency edges between each pair of words (w_t and w_j) where $b_{t,j}$=1 if there is an edge between them, and $b_{t,j}$=0 vice versa. There is also a dependency label matrix $\{r_{t,j}\}_{T\times T}$, where $r_{t,j}$ denotes the dependency relation label between w_t and w_j. We maintain the vectorial embedding $\boldsymbol{x}_{t,j}^r$ for each dependency label.

We denote the hidden representation of w_t at the l-th LAGCN layer as $\boldsymbol{e}_t^{s,l}$:

$$\boldsymbol{e}_t^{s,l} = \text{ReLU}(\textstyle\sum_{j=1}^{T} \alpha_{t,j}^l (\boldsymbol{W}_2 \cdot \boldsymbol{e}_j^{l-1} + \boldsymbol{W}_3 \cdot \boldsymbol{x}_{t,j}^r + \boldsymbol{W}_4 \cdot \boldsymbol{x}_j^p + b)), \tag{4}$$

where $\alpha_{t,j}^l$ is the syntactic-aware neighbor connecting-strength distribution calculated by:

$$\boldsymbol{r}_{t,j}^s = \boldsymbol{W}_5 \cdot [\boldsymbol{e}_j^{l-1}; \boldsymbol{x}_j^p; \boldsymbol{x}_{t,j}^r], \tag{5}$$

$$\alpha_{t,j}^l = \frac{b_{t,j} \cdot \exp(\boldsymbol{r}_{t,j}^s)}{\sum_{i=1}^{T} b_{t,i} \cdot \exp(\boldsymbol{r}_{t,i}^s)}, \tag{6}$$

where $\boldsymbol{r}_{t,j}^s$ entails the syntactic relationship between tokens. The weight distribution $\alpha_{t,j}$ entails the structural information, thus comprehensively reflecting the syntactic attributes.

We explicitly concatenate the representations of L-th layer of local attention POS encoder and LAGCN as the overall token representations $\boldsymbol{e}_t^L = [\boldsymbol{e}_t^{s,L}; \boldsymbol{e}_t^{p,L}]$.

2.3 Term Generation and Filtering

We concatenate BERT representation $\boldsymbol{v}_t$ and SynFue representation $\boldsymbol{e}_t^L$ as final token representation $\boldsymbol{h}_t$=$[\boldsymbol{v}_t; \boldsymbol{e}_t^L]$. We next construct span representation based on token representations:

$$\boldsymbol{h}_{pool} = \text{Max-Pooling}([\boldsymbol{h}_{head}, \cdots, \boldsymbol{h}_{tail}]), \tag{7}$$

$$\boldsymbol{s}_i^{'} = [\boldsymbol{h}_{head}; \boldsymbol{h}_{tail}; \boldsymbol{h}_S; \boldsymbol{h}_{size}; \boldsymbol{h}_{pool}], \tag{8}$$

$$\boldsymbol{s}_i = \text{FFN}(\text{Dropout}(\boldsymbol{s}_i^{'})), \tag{9}$$

where $\boldsymbol{h}_{head}$ and $\boldsymbol{h}_{tail}$ are the boundary representation of the start and end token of each term. $\boldsymbol{h}_S$ is the overall sentence representation from BERT (i.e., from *CLS* token). $\boldsymbol{h}_{pool}$ is the max pooling operation (Max-Pooling), and $\boldsymbol{h}_{size}$ is the term width embedding. 'FFN' refers to feed-forward layers, and 'Dropout' is applied to alleviate overfitting.

Then, we determine the term type via a softmax classifier $c_i = \text{Softmax}(\boldsymbol{s}_i)$. We pre-define three categories of terms: $\{C^A, C^O, C^\epsilon\}$, more specially, aspect term (C^A), opinion term (C^O) and invalid term (C^ϵ). Afterward, we estimate which type each term belongs to, by looking at the highest-scored class based on c_i, i.e., filtering invalid candidates (C^ϵ), maintaining a set of final terms which supposedly are aspect terms and opinion terms, i.e., a_n and o_m. Based on the filtering step, we can effectively prune the pool of span terms and lead to higher efficiency. We denote the representation of aspect term and opinion term as $\boldsymbol{s}_n^a$ and $\boldsymbol{s}_m^o$, respectively.

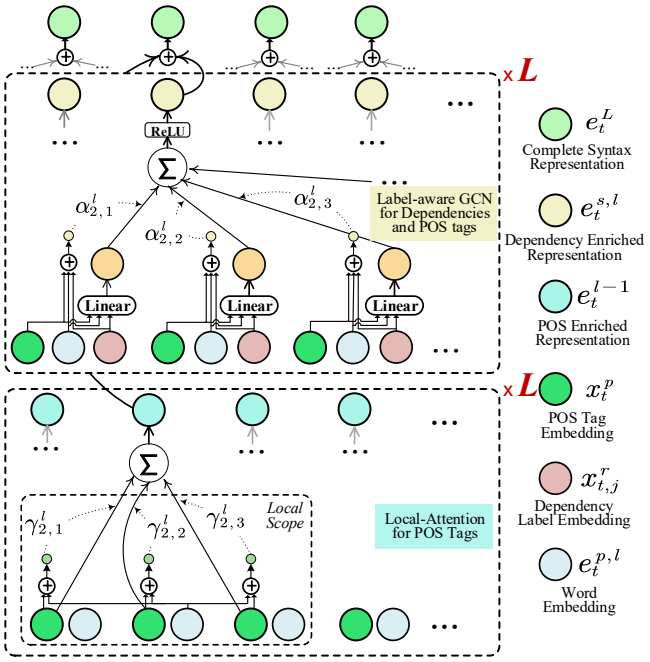

Figure 3: Illustration of the SynFue encoder.

2.4 Term Pairing

We measure the relationship between each candidate aspect-opinion term pair by calculating the potential scores of the term pairs $Q_{n,m} \in \mathbb{R}^{N\times M}$. We consider two types of scores: high-order pairing scores and syntactic-aware scores.

High-order scoring. In previous works, the Biaffine scorer is usually utilized for relation determination [Dozat and Manning, 2017; Eberts and Ulges, 2020]. However, Biaffine only models the dyadic relations between each pair, which may lead to an insufficient exploration of the triadic relations that occur in the overlapping aspect-opinion pair structures, as we highlighted earlier. We hence adopt high-order scoring, as in Figure 2. First, a Biaffine scorer is used for measuring each pair under the first-order scope:

$$\phi_{n,m}^{\texttt{Bi}} = \text{Sigmoid}(\begin{bmatrix} \mathbf{s}_n^a \\ 1 \end{bmatrix}^{\mathrm{T}} \mathbf{W}_6 \, \mathbf{s}_m^o), \tag{10}$$

where $\mathbf{W}_6 \in \mathbb{R}^{(d_s+1)\times d_s}$ is a parameter. d_s is the dimension of the term representations. The Triaffine scorer [Carreras, 2007; Wang *et al.*, 2019] is utilized for second-order scoring of two overlapping pairs simultaneously, over three term representations (i.e., $\mathbf{s}_n^a, \mathbf{s}_m^o, \mathbf{s}_k^*$)[2]

$$\phi_{n,m,k}^{\texttt{Tri}} = \text{Sigmoid}(\begin{bmatrix} \mathbf{s}_n^a \\ 1 \end{bmatrix}^{\mathrm{T}} (\mathbf{s}_m^o)^{\mathrm{T}} \mathbf{W}_7 \begin{bmatrix} \mathbf{s}_k^* \\ 1 \end{bmatrix}). \tag{11}$$

[2] $\mathbf{s}_k^*$ can be an aspect or opinion term, excluding $\mathbf{s}_n^a$ and $\mathbf{s}_m^o$.

Syntactic-aware scoring. Intuitively, the syntactic representations in LAGCN that depict the syntactic relationship between tokens can also provide rich clues for the detection of term pairs. Here we consider making use of such syntactic features, performing syntactic-aware scoring, cf. Figure 2. Technically, we re-harness the token-level representation $\boldsymbol{R}^s$=$[\cdots, \boldsymbol{r}^s_{t,j}, \cdots] \in \mathbb{R}^{T\times T}$ (from Eq. 5) by projecting $\boldsymbol{R}^s$ into span-level syntactic transition representations $\boldsymbol{S}^s \in \mathbb{R}^{N\times M}$. For an aspect-opinion term pair a_n and o_m, we first track its *start* and *end* indexes respectively in $\boldsymbol{R}^s$. We then obtain the transition representations $\boldsymbol{s}^p_{n,m}$, i.e., from a_n to o_m via the span pooling operation:

$$\boldsymbol{s}^p_{n,m} = \text{Span-Pooling}(\boldsymbol{r}^s_{n(start):n(end),m(start):m(end)}) . \tag{12}$$

We further apply a cross-attention operation [Ding *et al.*, 2020] for each pair $\boldsymbol{s}^p_{n,m}$, to propagate the dependencies and impacts from other terms at the same row and column. Concretely, we calculate the row-wise weights $\overleftrightarrow{\beta}$ and column-wise weights $\beta^{\updownarrow}$ on $\boldsymbol{s}^p_{n,m}$:

$$\overleftrightarrow{\beta}_k = \text{Softmax}(\frac{(\boldsymbol{s}^p_{n,m})^{\mathrm{T}} \cdot \boldsymbol{s}^p_{n,k}}{\sqrt{M}}), \beta^{\updownarrow}_k = \text{Softmax}(\frac{(\boldsymbol{s}^p_{n,m})^{\mathrm{T}} \cdot (\boldsymbol{s}^p_{k,m})}{\sqrt{N}}) , \tag{13}$$

where k is the column or row index of the current pair.

$$\phi^{\mathtt{S}}_{n,m} = \text{Sigmoid}(\boldsymbol{W}_8 \cdot (\textstyle\sum_k \overleftrightarrow{\beta}_k \boldsymbol{s}^p_{n,k} + \sum_k \beta^{\updownarrow}_k \boldsymbol{s}^p_{k,m})) . \tag{14}$$

Finally, we build the overall unary potential scores by taking into account all the above scoring items.

$$Q_{n,m} = \phi^{\mathtt{Bi}}_{n,m} + \eta_1 \sum_{k\neq n,m} \phi^{\mathtt{Tri}}_{n,m,k} + \eta_2 \, \phi^{\mathtt{S}}_{n,m} , \tag{15}$$

where η_1 and η_2 are factors regulating the contributions of different scores. We then push $Q_{n,m}$ into [0,1] likelihood value:

$$p_k(a_n, o_m) \leftarrow y_{n,m} = \text{Sigmoid}(Q_{n,m}) , \tag{16}$$

where those elements $y_{n,m}$ larger than a pre-defined threshold δ will be output as valid pairs, i.e., $p_k(a_n, o_m)$.

2.5 Training

During training, given an input sentence s with manually annotated gold pairs $\hat{P} = \{\hat{p}_k(\hat{a}, \hat{o})\}^K_{k=1}$. We define a joint loss for term detection and pair relation detection:

$$\mathcal{L} = \textstyle\sum^D (\mathcal{L}_{Type} + \lambda_1 \mathcal{L}_{Pair}) + \lambda_2 ||\theta||^2_2 , \tag{17}$$

where D is the total sentence number, λ_1 is the coupling co-efficiency regulating two loss items, and λ_2 is the ℓ_2 regularization factor. $\mathcal{L}_{Type}$ denotes the negative log-likelihood loss for term type detection, and $\mathcal{L}_{Pair}$ denotes the binary cross-entropy over pair relation classes:

$$\mathcal{L}_{Type} = -\textstyle\sum^G_{i=1} \hat{c}_i \log c_i , \tag{18}$$

$$\mathcal{L}_{Pair} = -\textstyle\sum^K_{k=1} \log p^{'}_k , \tag{19}$$

where G is total spans, $p^{'}_k$ is the factorized probability of each aspect-opinion pair over input sentence: $p^{'}_k = \prod_{a\in A, o\in O} p(a, o)$.

		#Sent.	#Asp.	#Opi.	#Pair	#Ovlp.P
14lap	Train	1,124	1,589	1,583	1,835	431 (23.49%)
	Test	332	467	478	547	147 (26.87%)
14res	Train	1,574	2,551	2,604	2,936	667 (22.72%)
	Test	493	851	866	1,008	276 (27.38%)
15res	Train	754	1,076	1,192	1,277	346 (27.09%)
	Test	325	436	469	493	98 (19.88%)
16res	Train	1,079	1,511	1,660	1,769	444 (25.10%)
	Test	328	456	485	525	120 (22.86%)

Table 1: Data statistics. '#Sent.', '#Asp.', '#Opi.' and '#Pair' denote the number of sentences, aspect/opinion terms and aspect-opinion pairs, respectively. '#Ovlp.P' is the number of overlapping pairs.

Negative sampling. During term type detection, in addition to the positive samples of the labeled terms, we randomly draw a fixed number (N_t) of negative samples, i.e., non-term spans, to accelerate the training.

3 Experiments

3.1 Experimental Setups

Datasets and resources. We conduct experiment on four benchmark datasets [Wu *et al.*, 2020a], including 14lap, 14res, 15res and 16res. The statistics of four datasets are listed in Table 1. We employ the Stanford CoreNLP Toolkit[3] to obtain the dependency parses and POS tags for all sentences. We adopt the officially released pre-trained BERT parameters.[4]

Implementation. Both the BERT representation v_t and the term span representation d_s have 768 dimensionality. The syntactic label embedding size and POS embedding are set to 100-d, and span width embedding is set to 25-d. We adopt the Adam optimizer with an initial learning rate of 4e-5. We use a batch size of 16 and set unfixed epochs with early-stop training strategy instead. We mainly adopt F1 score as the metric. Our model[5] takes different parameters on different data, which are separately fine-tuned.

Baselines. Our baselines are divided into pipeline methods and joint methods. • 1) One type of pipeline methods uses **CMLA** [Peng *et al.*, 2020] to co-extract aspect and opinion terms, and then make pairing with **CGCN** [Zhang *et al.*, 2018]. Another pipeline schemes first perform targeted aspect terms extraction, e.g., with **BiLSTM+ATT** [Fan *et al.*, 2018], **DECNN** [Xu *et al.*, 2018] and **RINANTE** [Dai and Song, 2019] models, and then conduct target-oriented opinion terms extraction with the given aspect terms in the second stage, e.g., by **IOG** [Fan *et al.*, 2019]. • 2) Joint methods perform unified extraction of aspect terms and opinion terms, as well as pair-wise relation between them, including **SpanMlt** [Zhao *et al.*, 2020] and **GTS** [Wu *et al.*, 2020a].

3.2 Results and Analysis

Main performances. The overall results are shown in Table 2. The first observation is that the performances by the joint methods are constantly higher than the two types of pipeline

[3]https://stanfordnlp.github.io/CoreNLP/, CoreNLP v4.2.0

[4]https://github.com/google-research/bert, base cased version.

[5]Available at https://github.com/ChocoWu/Synfue-PAOTE

	14lap	14res	15res	16res
• **Pipeline Methods**				
CMLA+CGCN†	53.03	63.17	55.76	62.70
BiLSTM+ATT+IOG†	52.84	65.46	57.73	64.13
DECNN+IOG†	55.35	68.55	58.04	64.55
RINANTE+IOG†	57.10	67.74	59.16	-
• **Joint Methods**				
SpanMlt†	64.41	73.80	59.91	67.72
GTS†	57.69	69.13	65.39	70.39
Ours w/o BERT♠	64.59	74.05	63.74	72.06
SpanMlt+BERT†	68.66	75.60	64.68	71.78
GTS+BERT†	65.67	75.53	67.53	74.62
Ours♠	**68.88**	**76.62**	**68.91**	**76.59**

Table 2: Main results. Baselines with the superscript '†' are copied from their raw papers; scores with '♠' are presented after a significant test with $p \leq 0.05$.

	14lap	14res	15res	16res	*Avg.*
Ours	**68.88**	**76.62**	**68.91**	**76.59**	**72.75**
• **Encoding**					
w/o BERT	64.59	74.05	63.74	72.06	68.08
w/o Dep.Label	68.67	76.13	68.52	76.42	72.44
GCN*	67.93	75.27	67.18	75.97	71.59
w/o LAGCN	66.33	75.41	64.54	74.31	70.15
w/o Loc.Att.	68.03	75.72	67.97	76.04	71.94
w/o POS tags	67.07	75.43	66.73	75.28	71.13
• **Decoding**					
w/o Neg.Samp.	67.56	73.72	68.28	76.12	71.42
w/o Biaffine (Eq. 10)	54.18	58.46	45.28	61.04	54.74
w/o Triaffine (Eq. 11)	68.20	76.02	67.77	76.14	72.03
w/o Syn.Score (Eq. 14)	67.58	75.87	67.01	75.18	71.41
w/o Cro.Att. (Eq. 13)	68.45	76.35	68.35	75.63	72.19

Table 3: Ablation results. 'GCN*' means replacing LAGCN with a vanilla GCN model that encodes only the syntactic dependency edges. 'w/o Dep.Label' means removing dependency labels from LAGCN while keeping dependency edges and POS tags.

methods. This confirms the previously established viewpoint that the joint scheme of aspect-opinion term extraction can relieve the error propagation issues in the pipeline. More importantly, our proposed model achieves the best results against all the baselines. For example, our model even without using BERT obtains 64.59%, 74.05%, 63.74% and 72.06% F1 scores on each dataset, respectively. By integrating the BERT language model, the performances of the joint models can be further improved. Note that our proposed model outperforms strong baselines by a large margin.

Ablation. We perform ablation experiments (cf. Table 3) to understand the effect of each part of the proposed model. We first remove BERT while using pre-trained Glove embeddings for BiLSTM instead, and we receive the most notable performance drops among all other factors, showing the effectiveness of BERT for downstream tasks [Zhao *et al.*, 2020; Wu *et al.*, 2020a; Fei *et al.*, 2020]. Without the dependency label features, we can find that the performances consistently decrease. By replacing LAGCN with vanilla GCN encoding only the dependency arcs, the results drop further. When LAGCN is ablated, i.e., without encoding the syntactic arcs and labels as well as the POS tags, such drops are magnified

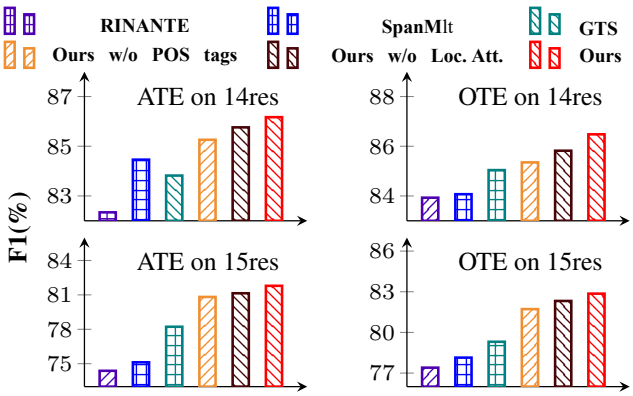

Figure 4: Results on extracting aspect and opinion terms on two datasets. Models all take the BERT representation.

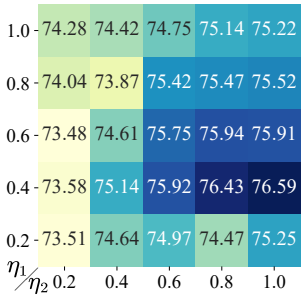

Figure 5: Pairing results by varying η_1 and η_2 on the 16res dataset.

significantly. If we strip off the local-attention POS encoder, or remove the POS features away from LAGCN, the performances are downgraded to some extent.

For decoding, we can find that the negative sampling strategy influences the results. Also, three pairing scores show effects in different extent, i.e., the first-order Biaffine gives more effects than the second-order Triaffine scorer and syntactic-aware scoring. One possible reason is that most non-overlapping pairs prevent Triaffine, which is more capable of modeling triadic relations among overlapping structures, from giving its utmost function. Furthermore, we can find that the syntactic-aware scores are highly crucial to the pairing, and removing the cross-attention mechanism will reduce the effectiveness of syntactic-aware scores.

Term extraction. We further examine our model's capability on aspect term extraction (ATE) and opinion term extraction (OTE), separately. Figure 4 shows the performances of two subtasks on the 14res and 15res datasets. It can be observed that the joint methods consistently outperform the pipeline method (RINANTE), while our model gives the best performances compared with all baselines. We also find that whether we use POS tagging features or not has a significant impact on term extraction.

Effects of pairing strategies. In addition to the base Biaffine scorer for term pairing, we further study how the Triaffine scorer and the syntactic-aware scoring influence the overall pairing performances. We reach this by tuning the regulating factors η_1 and η_2. From the patterns in Figure 5,

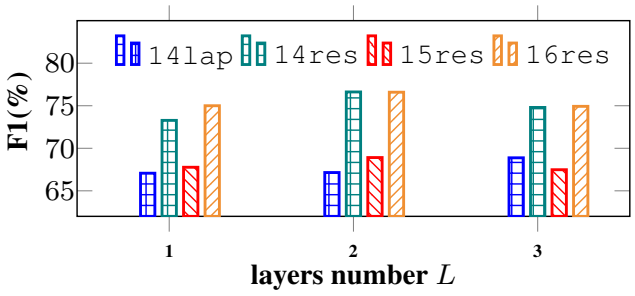

Figure 6: Performances under different layers of SynFue.

we can find that the overall result is the best when η_1=0.4 and η_2=1.0. For one thing, the comparatively fewer overlapping pairs in the dataset make the contribution by the Triaffine scorer limited. For another, the syntactic dependency information from LAGCN offers prominent hints for determining the semantic relations between aspect-opinion terms, which accordingly requires a higher proportion of scoring weights.

Influence of Layer Number. Syntactic fusion encoder (SynFue) is responsible for fusing syntactic structure features as well as the POS tags. Intuitively, more layers of SynFue should give stronger capability of the syntax modeling. We show the performances by installing different layers of SynFue in our model, based on each dataset. As illustrated in Figure 6, we see that the model can give the best effect with a two-layer of SynFue, in most of the datasets. This implies that too many layers of syntax propagation may partially result in information redundancy and overfitting.

Syntax correlations. Finally, we qualitatively investigate if our proposed LAGCN can genuinely model these syntaxes to improve the task. Technically, for each input sentence we observe the syntax-connecting weights α (in Eq. 6) and collect the weights of the correlated dependencies and POS tags of token words. We render these normalized values in Figure 7. It is quite clear to see that LAGCN well captures the correlations between syntactic dependencies and POS tags. For example, for the dependency arc with the '*nsubj*' type, LAGCN learns to assign more connections with those tokens with the POS tags of '*JJ*', '*NN*' and '*NNS*', which essentially depicts the boundary attributes of the constituent spans, as well as the correlated semantic relations between terms. This can explain the task improvement accordingly.

4 Related Work

Aspect terms extraction and opinion terms extraction, as two fundamental subtasks of fine-grained aspect-based sentiment analysis (ABSA) [Pang and Lee, 2007; Liu, 2012; Huang *et al.*, 2020; Wang *et al.*, 2020], have received extensive research attentions in recent years [Wang *et al.*, 2017; Xu *et al.*, 2018; Fan *et al.*, 2019; Chen and Qian, 2020]. Considering the relevance between two subtasks, Zhao et al. (2020) propose the pair-wise aspect and opinion terms extraction (PAOTE) task, detecting the pair of the correlated aspect and opinion terms jointly. Preliminary works adopt the pipeline methods, i.e., first extracting the aspect terms and the opinion terms

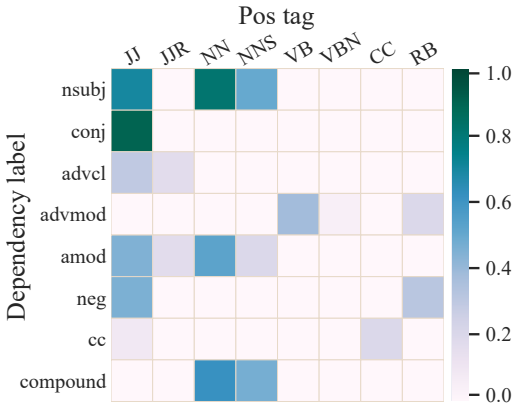

Figure 7: Correlations between syntactic dependencies and POS tags discovered by LAGCN. Only a high-frequency subset of syntactic labels are presented.

separately, and then making pairings for them [Wang *et al.*, 2017; Xu *et al.*, 2018; Peng *et al.*, 2020; Wu *et al.*, 2020b]. Recent efforts focus on designing joint extraction models for PAOTE [Wu *et al.*, 2020a; Chen *et al.*, 2020], reducing error propagation and bringing better task performances.

Previous studies also reveal that syntactic dependency features are crucial for ABSA [Phan and Ogunbona, 2020; Tang *et al.*, 2020]. These works mostly consider the syntactic dependency edges, while the syntactic labels and POS tags that also provide potential evidences, can not be exploited fully in the PAOTE task. We thus in this work propose a novel label-aware syntactic graph convolutional network for modeling rich syntactic features. Furthermore, we leverage the syntactic information for better term pairing. We also take advantage of the high-order graph-based models [Carreras, 2007; Wang *et al.*, 2019], i.e., using the second-order Triaffine scorer to fully explore the underlying mutual interactions within the overlapping pair structures.

5 Conclusions

In this study, we investigated a novel joint model for pair-wise aspect and opinion terms extraction (PAOTE). Our proposed syntax fusion encoder incorporated rich syntactic features, including dependency edges and labels, as well as the POS tags. During pairing, we considered both the high-order scoring and the syntactic-aware scoring for aspect-opinion term pairs. Experimental results on four benchmark datasets showed that our proposed syntax-enriched model gave improved performance compared with current state-of-the-art models, demonstrating the effectiveness of rich syntactic knowledge for this task.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 61772378), the National Key Research and Development Program of China (No. 2017YFC1200500), the Research Foundation of Ministry of Education of China (No. 18JZD015), and the Key Project of State Language Commission of China (No.ZDI135-112).

References

[Carreras, 2007] Xavier Carreras. Experiments with a higher-order projective dependency parser. In *EMNLP*, pages 957–961, 2007.

[Chen and Qian, 2020] Zhuang Chen and Tieyun Qian. Enhancing aspect term extraction with soft prototypes. In *EMNLP*, pages 2107–2117, 2020.

[Chen *et al.*, 2020] Shaowei Chen, Jie Liu, Yu Wang, Wenzheng Zhang, and Ziming Chi. Synchronous double-channel recurrent network for aspect-opinion pair extraction. In *ACL*, pages 6515–6524, 2020.

[Dai and Song, 2019] Hongliang Dai and Yangqiu Song. Neural aspect and opinion term extraction with mined rules as weak supervision. In *ACL*, pages 5268–5277, 2019.

[Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186, 2019.

[Ding *et al.*, 2020] Zixiang Ding, Rui Xia, and Jianfei Yu. ECPE-2D: emotion-cause pair extraction based on joint two-dimensional representation, interaction and prediction. In *ACL*, pages 3161–3170, 2020.

[Dozat and Manning, 2017] Timothy Dozat and Christopher D. Manning. Deep biaffine attention for neural dependency parsing. In *ICLR*, 2017.

[Eberts and Ulges, 2020] Markus Eberts and Adrian Ulges. Span-based joint entity and relation extraction with transformer pre-training. In *ECAI*, pages 2006–2013, 2020.

[Fan *et al.*, 2018] Feifan Fan, Yansong Feng, and Dongyan Zhao. Multi-grained attention network for aspect-level sentiment classification. In *EMNLP*, pages 3433–3442, 2018.

[Fan *et al.*, 2019] Zhifang Fan, Zhen Wu, Xin-Yu Dai, Shujian Huang, and Jiajun Chen. Target-oriented opinion words extraction with target-fused neural sequence labeling. In *NAACL*, pages 2509–2518, 2019.

[Fei *et al.*, 2020] Hao Fei, Meishan Zhang, and Donghong Ji. Cross-lingual semantic role labeling with high-quality translated training corpus. In *ACL*, pages 7014–7026, 2020.

[Huang *et al.*, 2020] Jiaxin Huang, Yu Meng, Fang Guo, Heng Ji, and Jiawei Han. Weakly-supervised aspect-based sentiment analysis via joint aspect-sentiment topic embedding. In *EMNLP*, pages 6989–6999, 2020.

[Liu, 2012] Bing Liu. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167, 2012.

[Luong *et al.*, 2015] Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *EMNLP*, pages 1412–1421, 2015.

[Marcheggiani and Titov, 2017] Diego Marcheggiani and Ivan Titov. Encoding sentences with graph convolutional networks for semantic role labeling. In *EMNLP*, pages 1506–1515, 2017.

[Nie *et al.*, 2020] Yuyang Nie, Yuanhe Tian, Yan Song, Xiang Ao, and Xiang Wan. Improving named entity recognition with attentive ensemble of syntactic information. In *EMNLP*, pages 4231–4245, 2020.

[Pang and Lee, 2007] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135, 2007.

[Peng *et al.*, 2020] Haiyun Peng, Lu Xu, Lidong Bing, Fei Huang, Wei Lu, and Luo Si. Knowing what, how and why: A near complete solution for aspect-based sentiment analysis. In *AAAI*, pages 8600–8607, 2020.

[Phan and Ogunbona, 2020] Minh Hieu Phan and Philip O. Ogunbona. Modelling context and syntactical features for aspect-based sentiment analysis. In *ACL*, pages 3211–3220, 2020.

[Tang *et al.*, 2020] Hao Tang, Donghong Ji, Chenliang Li, and Qiji Zhou. Dependency graph enhanced dual-transformer structure for aspect-based sentiment classification. In *ACL*, pages 6578–6588, 2020.

[Wang *et al.*, 2017] Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. Coupled multi-layer attentions for co-extraction of aspect and opinion terms. In *AAAI*, pages 3316–3322, 2017.

[Wang *et al.*, 2019] Xinyu Wang, Jingxian Huang, and Kewei Tu. Second-order semantic dependency parsing with end-to-end neural networks. In *ACL*, pages 4609–4618, 2019.

[Wang *et al.*, 2020] Kai Wang, Weizhou Shen, Yunyi Yang, Xiaojun Quan, and Rui Wang. Relational graph attention network for aspect-based sentiment analysis. In *ACL*, pages 3229–3238, 2020.

[Wu *et al.*, 2020a] Zhen Wu, Chengcan Ying, Fei Zhao, Zhifang Fan, Xinyu Dai, and Rui Xia. Grid tagging scheme for aspect-oriented fine-grained opinion extraction. In *EMNLP*, pages 2576–2585, 2020.

[Wu *et al.*, 2020b] Zhen Wu, Fei Zhao, Xin-Yu Dai, Shujian Huang, and Jiajun Chen. Latent opinions transfer network for target-oriented opinion words extraction. In *AAAI*, pages 9298–9305, 2020.

[Xu *et al.*, 2018] Hu Xu, Bing Liu, Lei Shu, and Philip S. Yu. Double embeddings and cnn-based sequence labeling for aspect extraction. In *ACL*, pages 592–598, 2018.

[Zhang *et al.*, 2018] Yuhao Zhang, Peng Qi, and Christopher D. Manning. Graph convolution over pruned dependency trees improves relation extraction. In *EMNLP*, pages 2205–2215, 2018.

[Zhang *et al.*, 2019] Chen Zhang, Qiuchi Li, and Dawei Song. Aspect-based sentiment classification with aspect-specific graph convolutional networks. In *EMNLP*, pages 4567–4577, 2019.

[Zhao *et al.*, 2020] He Zhao, Longtao Huang, Rong Zhang, Quan Lu, and Hui Xue. Spanmlt: A span-based multi-task learning framework for pair-wise aspect and opinion terms extraction. In *ACL*, pages 3239–3248, 2020.