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

Shengqiong Wu¹,*, Hao Fei¹,*, Yafeng Ren², Donghong Ji¹,† and Jingye Li²

¹Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan, China
²Guangdong 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) 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 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 overlapping¹ 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

¹An 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 propose for the first time in literature 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 \( s = \{w_1, \ldots, w_T\} \), our system is expected to produce a set of aspect-opinion pairs \( P = \{p_1(a, o), \ldots, p_k(a, o)\} \subset A \times O \).

\( A = \{a_1, \ldots, a_N\} \) is all possible aspect terms, where \( a_n \) can be a single word or a phrase, denoted as \( a_n = \{w_{i_1}, \ldots, w_{j_1}\} \). Likewise, \( O = \{o_1, \ldots, o_M\} \) is all possible opinion terms, where \( o_m = \{w_{i_2}, \ldots, w_{j_2}\} \) is a term span.

### 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:

\[
\{v_1, \ldots, v_T\} = \text{BERT}\left(\{w_1, \ldots, w_T\}\right),
\]

where \( v_k \) is the output representation for the word \( w_k \).

### 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 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 \( w_l \) in \( s \), we mark its corresponding POS tag as \( w_l^p \), and obtain its POS embedding \( \varphi_l \). At the \( l \)-th layer, the local attention operation is performed at a scope of \( d \) window size:

\[
\begin{align*}
e_{l}^{p,t} & = \sum_{i=t-d}^{t+d} \exp (W_1 e_i^{l-1}; \varphi_i^{p}), \\
\gamma_{l,i} & = \frac{\exp (W_1 e_i^{l-1}; \varphi_i^{p})}{\sum_{j=i-d}^{i+d} \exp (W_1 e_j^{l-1}; \varphi_j^{p})},
\end{align*}
\]

where \([;] \) denotes the concatenation operation, \( W_1 \) is the learnable parameters and \( e_{l}^{p,t} \) 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 $x_i^p$, we define an adjacency matrix $\{b_{i,j}\} _{T \times T}$ for dependency edges between each pair of words ($w_i$ and $w_j$) where $b_{i,j} = 1$ if there is an edge between them, and $b_{i,j} = 0$ vice versa. There is also a dependency label matrix $\{r_{i,j}\} _{T \times T}$, where $r_{i,j}$ denotes the dependency relation label between $w_i$ and $w_j$. We maintain the vectorial embedding $x_{i,j}^r$ for each dependency label.

We denote the hidden representation of $w_i$ at the $l$-th LAGCN layer as $e_i^l$:

$$e_i^l = \text{ReLU}(\sum_{j=1}^{T} \alpha_{i,j}^l (W_2 e_j^{l-1} + W_3 x_{i,j}^r + W_4 x_j^p + b)),$$

(4)

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

$$r_{i,j}^s = W_5 \cdot [e_j^{l-1}; x_j^p; x_{i,j}^r],$$

(5)

$$\alpha_{i,j}^l = \frac{b_{i,j} \cdot \exp(r_{i,j}^s)}{\sum_{i=1}^{T} b_{i,t} \cdot \exp(r_{i,t}^s)},$$

(6)

where $r_{i,j}^s$ entails the syntactic relationship between tokens. The weight distribution $\alpha_{i,j}^l$ 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 $e_i^L = [e_i^L; e_i^{L-1}]$.

2.3 Term Generation and Filtering

We concatenate BERT representation $v_i$ and SynFue representation $e_i^L$ as final token representation $h_i = [v_i; e_i^L]$. We next construct span representation based on token representations:

$$h_{pool} = \text{Max-Pooling}([h_{head}; \cdots; h_{tail}]^T),$$

(7)

$$s_i = [h_{head}; h_{tail}; h_S; h_{size}; h_{pool}],$$

(8)

$$s_i = \text{FFN}(\text{Dropout}(s_i)),$$

(9)

where $h_{head}$ and $h_{tail}$ are the boundary representation of the start and end token of each term. $h_S$ is the overall sentence representation from BERT (i.e., from CLS token). $h_{pool}$ is the max pooling operation (Max-Pooling), and $h_{size}$ 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 $c_i = \text{Softmax}(s_i)$. We pre-define three categories of terms: $\{C^A, C^O, C^*\}$, more specifically, aspect term ($C^A$), opinion term ($C^O$) and invalid term ($C^*$). 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 ($C^*$), maintaining a set of final terms which supposedly are aspect terms and opinion terms, i.e., $a_m$ 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 $s_{n,m}^a$ and $s_{n,m}^o$, respectively.

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. Hence we adopt high-order scoring, as in Figure 2. First, a Biaffine scorer is used for measuring each pair under the first-order scope:

$$\phi_{n,m}^{BI} = \text{Sigmoid}(\frac{1}{\theta} s_{n}^a \times \text{W}_6 \times s_{m}^o),$$. 

(10)

where $\text{W}_6 \in \mathbb{R}^{(d_s+1) \times d_s}$ 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., $s_{n,m}^a$, $s_{n,m}^o$, $s_{n,m}^k$):

$$\phi_{n,m,k}^{TR} = \text{Sigmoid}(\frac{1}{\theta} \text{W}_7 \times \text{S}_{o}^o \times \text{W}_6 \times s_{n,m,k}^a),$$

(11)

$^2 s_{n,m,k}^a$ can be an aspect or opinion term, excluding $s_{n,m}^a$ and $s_{n,m}^o$. 

Figure 3: Illustration of the SynFue encoder.
**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 $\mathbf{R}^s = \left[ \cdots, \mathbf{r}_i^s, \cdots \right] \in \mathbb{R}^{T \times T}$ (from Eq. 5) by projecting $\mathbf{R}^s$ into span-level syntactic transition representations $\mathbf{S}^s \in \mathbb{R}^{N \times M}$. For an aspect-opinion term pair $a_n$ and $o_m$, we first track its start and end indexes respectively in $\mathbf{R}^s$. Then we obtain the transition representations $\mathbf{s}_{n,m}^s$, i.e., from $a_n$ to $o_m$ via the span pooling operation:

$$
\mathbf{s}_{n,m}^s = \text{Span-Pooling}(\mathbf{r}_{\text{start}(n)}^s, m, \text{end}(m) | \mathbf{r}_{\text{end}(n)}^s).
$$

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

$$
\beta_k = \text{Softmax}\left( \frac{\mathbf{s}_{n,m}^s \cdot \mathbf{s}_{n,k}^p}{\sqrt{M}} \right), \beta'_k = \text{Softmax}\left( \frac{\mathbf{s}_{n,m}^s \cdot \mathbf{s}_{n,k}^p}{\sqrt{N}} \right),
$$

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

$$
\phi^s_{n,m} = \text{Sigmoid}(W_s \cdot (\sum_{k \neq n, m} \beta_k \mathbf{s}_{n,k}^p + \sum_{k \neq n, m} \beta'_k \mathbf{s}_{n,k}^p)).
$$

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

$$
Q_{n,m} = \phi_{n,m}^{\text{Ri}} + \eta_1 \sum_{k \neq n, m} \phi_{n,m,k}^{\text{Tri}} + \eta_2 \phi^s_{n,m},
$$

where $\eta_1$ and $\eta_2$ are factors regulating the contributions of different scores. We then push $Q_{n,m}$ into [0,1] likelihood value:

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

where those elements $\eta_{n,m}$ larger than a pre-defined threshold $\delta$ 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 $P = \{\mathbf{p}_k(a, o)\}_{k=1}^K$. We define a joint loss for term detection and pair relation detection:

$$
\mathcal{L} = \sum_{D} \left( \mathcal{L}_{\text{type}} + \lambda_1 \mathcal{L}_{\text{pair}} \right) + \lambda_2 \| \theta \|_2^2,
$$

where $D$ is the total sentence number, $\lambda_1$ is the coupling co-efficiency regulating two loss items, and $\lambda_2$ is the $\ell_2$ regularization factor. $\mathcal{L}_{\text{type}}$ denotes the negative log-likelihood loss for term type detection, and $\mathcal{L}_{\text{pair}}$ denotes the binary cross-entropy over pair relation classes:

$$
\mathcal{L}_{\text{type}} = - \sum_{c=1}^{G} c_i \log c_i, \quad \mathcal{L}_{\text{pair}} = - \sum_{k=1}^{K} \log p'_{k},
$$

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

### 3 Experiments

#### 3.1 Experimental Setups

**Datasets and resources.** We conduct experiment on four benchmark datasets [Wu et al., 2020], including 14lap, 14res, 15res and 16res. The statistics of four datasets are listed in Table 1. We employ the Stanford CoreNLP Toolkit\footnote{https://stanfordnlp.github.io/CoreNLP/} to obtain the dependency parses and POS tags for all sentences. We adopt the officially released pre-trained BERT parameters.\footnote{https://github.com/google-research/bert, base cased version.}

**Implementation.** Both the BERT representation $v_t$ and the term span representation $d_t$ 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\footnote{Available at https://github.com/ChocoWu/Synfue-PAOTE} takes different parameters on different data, which are separately fine-tuned.

**Baselines.** Our baselines are divided into pipeline methods and joint methods. \(\bullet\) 1) One type of pipeline methods uses CMLA [Peng et al., 2020] to co-extract aspect and opinion terms, and then make pairing with GCNC [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 RIANTE [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]. \(\bullet\) 2) Joint methods perform unified extraction of aspect terms and opinion terms, as well as pair-wise relation between them, including SpanMit [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

<table>
<thead>
<tr>
<th>#Sent.</th>
<th>#Asp.</th>
<th>#Opt.</th>
<th>#Pair</th>
<th>#Ovlp.P</th>
</tr>
</thead>
<tbody>
<tr>
<td>14lap</td>
<td>1,124</td>
<td>1,589</td>
<td>1,583</td>
<td>1,835</td>
</tr>
<tr>
<td>14res</td>
<td>332</td>
<td>467</td>
<td>478</td>
<td>547</td>
</tr>
<tr>
<td>15res</td>
<td>493</td>
<td>851</td>
<td>866</td>
<td>1,008</td>
</tr>
<tr>
<td>16res</td>
<td>325</td>
<td>436</td>
<td>469</td>
<td>493</td>
</tr>
<tr>
<td></td>
<td>Test</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14lap</td>
<td>328</td>
<td>456</td>
<td>485</td>
<td>525</td>
</tr>
</tbody>
</table>

Table 1: Data statistics. `

**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.
Table 2: Main results. Baselines with the superscript ‘\(^\dagger\)’ are copied from their raw papers; scores with ‘\(\star\)’ are presented after a significant test with \(p \leq 0.05\).

<table>
<thead>
<tr>
<th>Pipeline Methods</th>
<th>14lap</th>
<th>14res</th>
<th>15res</th>
<th>16res</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMLA+GCN(^\dagger)</td>
<td>53.03</td>
<td>63.17</td>
<td>55.76</td>
<td>62.70</td>
</tr>
<tr>
<td>BiLSTM+ATT+IOG(^\dagger)</td>
<td>52.84</td>
<td>65.46</td>
<td>57.73</td>
<td>64.13</td>
</tr>
<tr>
<td>DECN+IOG(^\dagger)</td>
<td>55.35</td>
<td>68.55</td>
<td>58.04</td>
<td>64.55</td>
</tr>
<tr>
<td>RINANTE+IOG(^\dagger)</td>
<td>57.10</td>
<td>67.74</td>
<td>59.16</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Joint Methods</th>
<th>14lap</th>
<th>14res</th>
<th>15res</th>
<th>16res</th>
</tr>
</thead>
<tbody>
<tr>
<td>SpanM(^\dagger)</td>
<td>64.41</td>
<td>73.80</td>
<td>59.91</td>
<td>67.72</td>
</tr>
<tr>
<td>GTS(^\dagger)</td>
<td>57.69</td>
<td>69.13</td>
<td>65.39</td>
<td>70.39</td>
</tr>
<tr>
<td>Ours w/o POS tags</td>
<td>65.07</td>
<td>76.62</td>
<td>68.91</td>
<td>76.59</td>
</tr>
<tr>
<td>Ours w/o Dep.Label</td>
<td>64.59</td>
<td>74.05</td>
<td>63.74</td>
<td>72.06</td>
</tr>
<tr>
<td>Ours w/o LAGCN</td>
<td>68.66</td>
<td>75.60</td>
<td>64.68</td>
<td>71.78</td>
</tr>
<tr>
<td>Ours w/o BER</td>
<td>65.67</td>
<td>75.53</td>
<td>67.53</td>
<td>74.62</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Encoding</th>
<th>14lap</th>
<th>14res</th>
<th>15res</th>
<th>16res</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours</td>
<td>68.88</td>
<td>76.62</td>
<td>68.91</td>
<td>76.59</td>
<td>72.75</td>
</tr>
<tr>
<td>w/o BERT</td>
<td>64.59</td>
<td>74.05</td>
<td>63.74</td>
<td>72.06</td>
<td>68.08</td>
</tr>
<tr>
<td>w/o Dep.Label</td>
<td>68.57</td>
<td>76.33</td>
<td>65.52</td>
<td>74.32</td>
<td>72.74</td>
</tr>
<tr>
<td>GCN*</td>
<td>67.93</td>
<td>75.27</td>
<td>71.8</td>
<td>75.97</td>
<td>71.59</td>
</tr>
<tr>
<td>w/o LAGCN</td>
<td>66.33</td>
<td>75.41</td>
<td>64.54</td>
<td>74.31</td>
<td>70.15</td>
</tr>
<tr>
<td>w/o Loc.Att.</td>
<td>68.03</td>
<td>75.72</td>
<td>76.04</td>
<td>71.34</td>
<td>71.13</td>
</tr>
<tr>
<td>w/o POS tags</td>
<td>67.07</td>
<td>75.43</td>
<td>76.76</td>
<td>71.28</td>
<td>71.13</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Decoding</th>
<th>14lap</th>
<th>14res</th>
<th>15res</th>
<th>16res</th>
</tr>
</thead>
<tbody>
<tr>
<td>w/o Neg.Samp.</td>
<td>67.56</td>
<td>73.72</td>
<td>68.28</td>
<td>76.12</td>
</tr>
<tr>
<td>w/o BiAffine (Eq. 10)</td>
<td>54.18</td>
<td>58.46</td>
<td>54.28</td>
<td>61.04</td>
</tr>
<tr>
<td>w/o TriAffine (Eq. 11)</td>
<td>68.20</td>
<td>76.02</td>
<td>76.77</td>
<td>76.14</td>
</tr>
<tr>
<td>w/o Syn.Score (Eq. 14)</td>
<td>67.58</td>
<td>75.87</td>
<td>76.01</td>
<td>71.18</td>
</tr>
<tr>
<td>w/o Cro.Att. (Eq. 13)</td>
<td>68.45</td>
<td>76.35</td>
<td>68.35</td>
<td>75.63</td>
</tr>
</tbody>
</table>

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.

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 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 Bi-affine scorer for term pairing, we further study how the Tri-affine scorer and the syntactic-aware scoring influence the overall pairing performances. We reach this by tuning the regulating factors \(\eta_1\) and \(\eta_2\). From the patterns in Figure 5,
we can find that the overall result is the best when $\eta_1=0.4$ and $\eta_2=1.0$. 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 quantitatively 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 $\alpha$ (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 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


