Learning Cooperative Interactions for Multi-Overlap Aspect Sentiment Triplet Extraction

ShimanZhao\textsuperscript{1,2}, WeiChen\textsuperscript{1,2,3} and TengjiaoWang\textsuperscript{1,2,3}

\textsuperscript{1} School of Computer Science, Peking University, Beijing, China
\textsuperscript{2} Research Center for Computational Social Science, Peking University
\textsuperscript{3} Institute of Computational Social Science, Peking University (Qingdao)
shimanzhao@stu.pku.edu.cn, \{pekingchenwei, tjwang\}@pku.edu.cn

Abstract

Aspect sentiment triplet extraction (ASTE) is an essential task, which aims to extract triplets (aspect, opinion, sentiment). However, overlapped triplets, especially multi-overlap triplets, make ASTE a challenge. Most existing methods suffer from multi-overlap triplets because they focus on the single interactions between an aspect and an opinion. To solve the above issues, we propose a novel multi-overlap triplet extraction method, which decodes the complex relations between multiple aspects and opinions by learning their cooperative interactions. Overall, the method is based on an encoder-decoder architecture. During decoding, we design a joint decoding mechanism, which employs a multi-channel strategy to generate aspects and opinions through the cooperative interactions between them jointly. Furthermore, we construct a correlation-enhanced network to reinforce the interactions between related aspects and opinions for sentiment prediction. Besides, a relation-wise calibration scheme is adopted to further improve performance. Experiments show that our method outperforms baselines, especially multi-overlap triplets.

1 Introduction

Aspect-based sentiment analysis (ABSA) (Pontiki et al., 2016) is very important for natural language processing (Atkinson-Abutridy, 2022), which involves several fine-grained information extraction tasks (Liang et al., 2020; Zhang et al., 2020b; Wan et al., 2020; Wang et al., 2021; Liu et al., 2021; Zhang et al., 2021; Hu et al., 2021). Recently, aspect sentiment triplet extraction (ASTE) as an essential ABSA task has received widespread attention, which aims to extract triplets consisting of an aspect, an opinion, and their corresponding sentiment. Generally, there are some overlapped triplets, especially multi-overlap triplets, due to

Unstructured text:
The ease of use and the top (slightly expensive) service from Apple never disappoint.

Aspect sentiment triplets:
\begin{itemize}
\item \text{use}, \text{ease}, \text{positive}
\item \text{use}, \text{never disappoint}, \text{positive}
\item \text{service}, \text{never disappoint}, \text{positive}
\item \text{service}, \text{slightly expensive}, \text{negative}
\item \text{service}, \text{top}, \text{positive}
\end{itemize}

Table 1: Overlapped triplets under multi-aspect and multi-opinion. And multi-overlap triplets are shown in the dotted box. Besides, aspects, opinions, and sentiments are marked with red, blue, and green.

the complex relations between aspects and opinions. Table 1 shows the example, “The ease of use and the top (slightly expensive) service from Apple never disappoint”. Obviously, the aspect “use” has two opinions “ease” and “never disappoint”, whereas the other aspect “service” has three opinions “top”, “slightly expensive” and “never disappoint”. And the opinion “never disappoint” is shared by “use” and “service”. The multi-overlap triplets are shown in Table 1 dotted box, and they have an overlapped aspect and an overlapped opinion. Obviously, multi-overlap triplets are more challenging than other overlapped triplets in capturing the relations between aspects and opinions. Most existing methods suffer from multi-overlap triplets. Thus, they cannot fully solve ASTE task.

There are two major research lines on ASTE task: tag-aware methods and span-aware methods. Tag-aware methods utilize tagging schemes to identify three factors of a triplet. However, most of them (Peng et al., 2020; Xu et al., 2020) cannot address the words with multi-tag by assigning a fixed tag to a word. Therefore, they fail to extract multi-overlap triplets. Besides, these methods (Wu et al., 2020; Chen et al., 2021b) suffer from triplets with multi-word spans because they focus on the interactions
between words. Span-aware methods are free from the trouble of multi-word spans because they consider the whole spans to identify the start and end boundaries. Span-aware methods mainly include question-driven (Mao et al., 2021; Gao et al., 2021) and joint generation (Yan et al., 2021; Mukherjee et al., 2021). However, these methods also fail to solve multi-overlap triplets because they focus on the single interactions between an aspect and an opinion when extracting triplets.

In short, most existing methods are plagued with multi-overlap triplets. In this paper, we propose an effective multi-overlap triplet extraction method, which decodes the complex relations between multiple aspects and opinions by learning their cooperative interactions. Overall, we adopt an encoder-decoder architecture. And a joint decoding mechanism (JDM) is designed in the decoding process: employing a multi-channel strategy to learn cooperative interactions between multiple aspects and opinions and promote their generation in different channels. Furthermore, we construct a correlation-enhanced network (CEN) by encoding the context with dependency relations, reinforcing the interactions between related aspects and opinions when predicting their sentiments. Besides, we design a relation-wise calibration scheme to filter out unfaithful triplets and alleviate error propagation. Our method can effectively solve overlapped triplets, especially multi-overlap triplets.

Our contributions are summarized as follows:

• We propose a multi-overlap triplet extraction method, which decodes the complex relations between aspects and opinions by learning their cooperative interactions. Our method can effectively solve multi-overlap triplets.

• We design a joint decoding mechanism, which employs a multi-channel strategy to capture the cooperative interactions between multiple aspects and opinions and promote their generations in different channels.

• We construct a correlation-enhanced network to enhance the interactions between related aspects and opinions when predicting their sentiments.

• Extensive experiments show that our method outperforms baselines. Besides, it achieves significant improvement for multi-overlap triplets.

2 Related work

The ASTE task includes three fundamental tasks: aspect term extraction (Xu et al., 2018), opinion term extraction (Yu et al., 2018), and aspect-oriented sentiment classification (Pontiki et al., 2016). These fundamental tasks are point keys in solving ASTE task. As a compound task, ASTE has two main research lines, including tag-aware methods (Xu et al., 2020; Zhang et al., 2020a; Chen et al., 2021b) and span-aware methods (Chen et al., 2021a; Mao et al., 2021; Xu et al., 2021). Then, we introduce related works on each research line.

Tag-aware methods assign a single tag to each word by tagging schemes. Peng et al. (2020) utilize two BIEOS-based sequence tagging schemes to extract aspect-sentiment pairs and opinions and then identify their relations. Xu et al. (2020) leverage a unified sequence tagging scheme to jointly extract three factors of a triplet. Chen et al. (2021b) propose a grid tagging scheme to tag relations between word-word pairs to fill a sentiment relation table. However, these methods limit the interactions between aspects and opinions by assigning a fixed tag to each word, and they ignore the impact of the relations between multiple aspects and opinions. Therefore, they fail to solve multi-overlap triplets.

Span-aware methods identify the aspect and opinion spans by considering the start and end boundaries. Mao et al. (2021) propose a question-driven method based on a reading comprehension scheme. They select one or more answers to a question to extract three factors of a triplet. Then, Chen et al. (2021a) propose a bidirectional question-driven method to solve ASTE task. However, these methods focus on the interactions between a question and its answers. Therefore, they still cannot fully solve multi-overlap triplets. Mukherjee et al. (2021) propose a generation method based on a recurrent neural network (RNN) to decode entire triplets. However, they also cannot fully solve multi-overlap triplets due to the limitation of the single interactions between an aspect and an opinion. Yan et al. (2021) propose a unified generation framework to extract triplets through a sequence output. However, they still suffer from multi-overlap triplets.

Unlike the above methods, our method considers the complex relations between aspects and opinions and captures their cooperative interactions to solve multi-overlap triplets.
3 Method

3.1 Task Formulation

For ASTE task, given a sentence $X = \{x_1, x_2, ..., x_n\}$. The sentence $X$ is a sequence of words, $x_i$ is the $i^{th}$ word, and $n$ is the length of the sentence. We use $a$, $o$, and $s$ to represent aspect spans, opinion spans, and sentiment polarities, respectively. Besides, the superscript $s$ and $e$ denote the start position and the end position of a span. Therefore, we formulate that $(a^s, a^e)$ denotes the span of an aspect and $(o^s, o^e)$ denotes the span of an opinion. Besides, $s^p$ denotes sentiment polarities, where $p \in \{\text{Positive, Neutral, Negative}\}$. Each aspect sentiment triplet is defined as a 5-point tuple $T^i = (a^s_i, a^e_i, o^s_i, o^e_i, s^p_i)$. ASTE task aims to extract all aspect sentiment triplets in a text.

3.2 Model Architecture

As shown in Figure 1, our method consists of a representation encoder (RE), joint decoding mechanism (JDM), and correlation-enhanced network (CEN) components. Specially, three components are depicted as follows:

**RE**. The RE component takes the Bart-encoder as a backbone and then encodes the sentence to obtain contextualized hidden representations.

**JDM**. The JDM component consists of a sharing decoder unit with a pointer-network and three channels (i.e., an aspect channel, an opinion channel, and an auxiliary channel). The sharing decoder unit takes the Bart-decoder as a backbone and then replicates the same structure to these three channels so that they can share parameters during training. And the JDM component can jointly train and optimize these three channels through the sharing decoder unit and generate some candidate aspect and opinion spans.

**CEN**. The CEN component consists of an interaction enhancement module and a relation-wise calibration scheme. The interaction enhancement module encodes the context with dependency relations to reinforce the interactions between related aspects and opinions while predicting their sentiments. And the error propagation is alleviated by the relation-wise calibration scheme.

3.2.1 The RE component

For a sentence $X = \{x_1, x_2, ..., x_n\}$, we add a special start token $[CLS]$ and a special end token $[SEP]$ to $X$, which constructs "$[CLS] X [SEP]$" as an input to the RE component. Then, the RE component encodes the input into hidden states $H^E$. And we define $H^E = [h^E_{CLS}, h^E_X, h^E_{SEP}]$, where $h^E_X = [h^E_{x_1}, h^E_{x_2}, ..., h^E_{x_n}]$. The $H^E$ is as follows:

$$H^E = RE([CLS], x_1, x_2, ..., x_n, [SEP])$$

where $H^E \in \mathbb{R}^{n \times d}$, and $d$ is the hidden dimension.
3.2.2 The JDM component
We divide the whole sentence into aspect spans, opinion spans, and other spans. Besides, we use \((\text{aux}^a, \text{aux}^c)\) to represent other spans. Then, we construct target sequences for three channels.

To identify different channels, we add channel symbol tokens \([\text{AC}], [\text{OC}], \text{and } [\text{AuxC}]\) to different target sequences. And, a special start token \([s]\) and a special end token \([/s]\) are added to target sequences. The target sequences are as follows:

- **Aspect Channel:**
  \[
  Y = [s], [\text{AC}], a^s_1, a^c_1, \ldots, a^s_{T}, a^c_{T}, [/s]
  \]

- **Opinion Channel:**
  \[
  Y = [s], [\text{OC}], o^a_1, o^c_1, \ldots, o^a_{T}, o^c_{T}, [/s]
  \]

- **Auxiliary Channel:**
  \[
  Y = [s], [\text{AuxC}], \text{aux}^a_1, \text{aux}^c_1, \ldots, \text{aux}^a_{T}, \text{aux}^c_{T}, [/s]
  \]

The target sequence of each channel consists of pointer indexes, which refer to the position indexes of a sentence. Figure 2 presents a sentence with pointer indexes and the examples of three channels.

Each channel takes the hidden states \(H^E\) of the RE component and previous outputs \(\hat{Y}_{<t}\) of the channel as inputs to get the next hidden state \(h^D_t\).

\[
\hat{h}^D_t = \text{Decoder}(H^E, \hat{Y}_{<t})
\]

where \(h^D_t \in R^d\), the probability distribution \(P_t\) is as follows.

\[
H^{E\prime} = \text{MLP}(H^E)
\]

\[
M^D = \text{Decoder}.\text{embed\_tokens}(X)
\]

\[
M^{D\prime} = \text{Dropout}(M^D)
\]

\[
P_t = \text{Softmax}\left(\frac{1}{2}(H^{E\prime} + M^{D\prime})\hat{h}^D_t\right)
\]

where \(H^E, H^{E\prime}, M^D, M^{D\prime} \in R^{n \times d}, h^D_t \in R^{1 \times d}\), and \(P_t \in R^n\) is the probability distribution on word indexes from a sentence.

During training, we define the target sequence of the aspect channel as \(Y^a = \{y^a_1, y^a_2, \ldots, y^a_T\}\), the target sequence of the opinion channel as \(Y^o = \{y^o_1, y^o_2, \ldots, y^o_T\}\), and the target sequence of the auxiliary channel as \(Y^{aux} = \{y^{aux}_1, y^{aux}_2, \ldots, y^{aux}_T\}\). Then, we utilize a cross-entropy for optimization with \(L_{JDM}\):

\[
L_{JDM} = - \sum_{t=1}^{T_a} \left( \sum_{t=1}^{T_a} y^a_t \log(\hat{y}^a_t) + \sum_{t=1}^{T_o} y^o_t \log(\hat{y}^o_t) \right)
\]

\[
+ \sum_{t=1}^{T_{aux}} y^{aux}_t \log(\hat{y}^{aux}_t)
\]

During inference, we put the start token \([s]\) and the channel symbol token (i.e., \([\text{AC}], [\text{OC}],\) or \([\text{AuxC}]\)) instead of a single \([s]\) into different channels to decode the first token of the target sequences. Besides, we use beam search to get output sequences \(\hat{Y}\) in an autoregressive manner.

3.2.3 The CEN component
The CEN component aims to identify the relations between aspects and opinions for multi-overlap triplets. It includes an interaction enhancement module and a relation-wise calibration scheme. First, the interaction enhancement module enhances the interactions between related aspects and opinions for sentiment prediction by encoding context dependency. And then, a relation-wise calibration scheme is adopted to make error rectification. The detailed descriptions are as follows.

The CEN component takes the hidden states \(H^E\) and the output sequences \(\hat{Y}^a\) and \(\hat{Y}^o\) as inputs. We convert \(\hat{Y}^a\) and \(\hat{Y}^o\) into aspect and opinion spans.
Datasets

<table>
<thead>
<tr>
<th></th>
<th>D_{20a} triplets in a sentence</th>
<th>D_{20b} triplets in a sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#S</td>
<td>#MulPol</td>
</tr>
<tr>
<td>D20a</td>
<td></td>
<td></td>
</tr>
<tr>
<td>train</td>
<td>1300</td>
<td>107</td>
</tr>
<tr>
<td>test</td>
<td>323</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>496</td>
<td>28</td>
</tr>
<tr>
<td>D20b</td>
<td></td>
<td></td>
</tr>
<tr>
<td>train</td>
<td>920</td>
<td>37</td>
</tr>
<tr>
<td>dev</td>
<td>228</td>
<td>5</td>
</tr>
<tr>
<td>test</td>
<td>339</td>
<td>14</td>
</tr>
</tbody>
</table>

Table 2: The sentence-level statistics of four datasets on D_{20a} and D_{20b}. #S denotes the overall number of sentences. #MulPol denotes the number of sentences with triplets of different sentiments. #OverLap denotes the number of sentences with overlapped triplets. And the items (i.e., 1, 2, 3, 4, and ≥ 5) denote the number of sentences with 1, 2, 3, 4, and more triples.

<table>
<thead>
<tr>
<th></th>
<th>D_{20a}</th>
<th>D_{20b}</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#S</td>
<td>#T</td>
</tr>
<tr>
<td>train</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D20a</td>
<td></td>
<td></td>
</tr>
<tr>
<td>train</td>
<td>1300</td>
<td>2145</td>
</tr>
<tr>
<td>dev</td>
<td>323</td>
<td>524</td>
</tr>
<tr>
<td>test</td>
<td>496</td>
<td>862</td>
</tr>
<tr>
<td>train</td>
<td>920</td>
<td>1265</td>
</tr>
<tr>
<td>dev</td>
<td>228</td>
<td>337</td>
</tr>
<tr>
<td>test</td>
<td>339</td>
<td>490</td>
</tr>
<tr>
<td>train</td>
<td>593</td>
<td>923</td>
</tr>
<tr>
<td>dev</td>
<td>148</td>
<td>238</td>
</tr>
<tr>
<td>test</td>
<td>318</td>
<td>455</td>
</tr>
<tr>
<td>train</td>
<td>842</td>
<td>1289</td>
</tr>
<tr>
<td>dev</td>
<td>210</td>
<td>316</td>
</tr>
<tr>
<td>test</td>
<td>320</td>
<td>465</td>
</tr>
</tbody>
</table>

Table 3: The triplet-level statistics of four datasets on D_{20a} and D_{20b}, where #T denotes the number of triplets, ‘+’, ‘0’, and ‘-’ denote the number of positive, neutral, and negative sentiments. And #AOT and #OOT denote the number of aspect-overlapped and opinion-overlapped triplets.

**Interaction Enhancement**

We utilize a graph convolutional network (GCN) to enhance the interactions between related aspects and opinions. First, we obtain dependency relations for sentiment prediction. It is crucial to solving multi-overlap triplets.

Then, we concatenate the start word representations of an aspect span and an opinion span. Furthermore, we get enhanced states \( H = [h_1^G, h_2^G, ...] \). Then, we concatenate the start word representations of the aspect and opinion spans to predict their sentiments. The equations are as follows.

\[
sr_{ao}^G = [h_{ao}^T, h_{ao}^G]
\]

\[
p(m|sr_{ao}^G) = \text{Softmax}(f(W_m^1sr_{ao}^G + b_m^1)W_m^2 + b_m^2)
\]

where \( sr_{ao}^G \) denotes the concatenation of the start word representations of an aspect span and an opinion span. Furthermore, \( W_m^1, b_m^1, W_m^2, \) and \( b_m^2 \) are learnable parameters. And \( m \in \{\text{Positive, Negative, Neutral, None}\} \).

GCN is valuable to the dependency graph, but it heavily relies on the qualities of the parser tree. In this component, a relation-wise calibration scheme is adopted to alleviate dependency parser errors. Therefore, GCN can effectively reinforce the interactions between related aspects and opinions for sentiment prediction. It is crucial to solving multi-overlap triplets.
Relation-wise Calibration

We can obtain the representations of aspect spans and opinion spans based on $H^E$. Every word can not play an equal role in a multi-word span representation. Therefore, we use self-attention to convert a multi-word span representation into a vector, emphasizing the meaning of important words. We formulate that $sr$ denotes the span representation. The $sr$ is described as follows:

$$M_{sr} = [h_i^E : h_j^E] \quad \text{(11)}$$

$$A_{sr} = \text{Softmax}(W_{sr}^1 \tanh(W_{sr}^2 M_{sr}^T)) \quad \text{(12)}$$

$$sr = A_{sr} M_{sr} \quad \text{(13)}$$

where $M_{sr} \in R^{L \times d}$ denotes all word representations of the span from start position $i$ to end position $j$, and $L$ denotes the number of words in the span. $A_{sr} \in R^L$ is a weight matrix, and $sr \in R^d$. $W_{sr}^1$ and $W_{sr}^2$ are learnable parameters.

We formulate that $sr_a$ and $sr_o$ denote the representations of the aspect span and the opinion span. Then, we perform Cartesian Product on aspect spans and opinion spans to obtain candidate aspect-opinion pair representations. The candidate set $SR_{ao} = \{sr_{ao}^1, sr_{ao}^2, ..., sr_{ao}^q\}$, where $sr_{ao}^q$ denotes the $q^{th}$ aspect-opinion pair representation:

$$sr_{ao}^q = [sr_{a}^i, sr_{o}^j, sr_{a}^i - sr_{o}^j] \quad \text{(14)}$$

where $sr_{a}^i$ denotes the $i^{th}$ representation of the aspect span, $sr_{o}^j$ denotes the $j^{th}$ representation of the opinion span, and $sr_{a}^i - sr_{o}^j$ denotes the difference value between them.

Then, we use two linear layers to identify related aspect-opinion pairs from the candidate set as a calibration scheme to identify unfaithful triplets. The equation is as follows:

$$p(r|sr_{ao}) = \text{Sigmoid}(f(W_{ao}^1 sr_{ao} + b_{ao}^1)W_{ao}^2 + b_{ao}^2)) \quad \text{(15)}$$

where $W_{ao}^1$, $W_{ao}^2$, $b_{ao}^1$, and $b_{ao}^2$ are learnable parameters, $f(.)$ denotes a non-linear activation function, and $r \in \{\text{Valid, Invalid}\}$.

We utilize log-likelihood to optimize $L_{CEN}$:

$$L_{CEN} = - \sum \sum \log P(r^*|sr_{ao}) + \sum \log P(m^*|sr_{ao}^G) \quad \text{(16)}$$

where $r^*$ is a validation between positive and negative instances, and $m^*$ is the sentiment relation between an aspect and an opinion.

3.2.4 Training

The training objective is the loss sum of the JDM component and the CEN component.

$$L = \alpha L_{JDM} + \beta L_{CEN} \quad \text{(17)}$$

where $\alpha$ and $\beta$ are hyperparameters.

4 Experiment

4.1 Datasets

We evaluate our method on $D_{20a}$ dataset (Peng et al., 2020) and $D_{20b}$ dataset (Xu et al., 2020). These two datasets include three sub-datasets (i.e., 14res, 15res, 16res) in a restaurant domain and a sub-dataset (i.e., 14lap) in a laptop domain. $D_{20a}$ is the revised version of $D_{20a}$, including more overlapped triplets. Specially, we calculate the number of sentences with overlapped triplets, that is $D_{20a}$ is 21.00%, 17.62%, 19.26%, 19.02% and $D_{20b}$ is 32.54%, 29.66%, 25.86%, 25.48% on 14res, 14lap, 15res, 16res, respectively. We give the sentence-level and triplet-level statistics on $D_{20a}$ and $D_{20b}$. The detailed statistics are presented in Table 2 and 3. The datasets and codes are available 1.

4.2 Baselines

The baselines can be summarized as two groups: tag-aware methods and span-aware methods.

Tag-aware methods. RIANTE model (Dai and Song, 2019) and Li-unified model (Li et al., 2019) use sequence tagging schemes to solve aspect-opinion pair extraction task and aspect-sentiment pair extraction task. Peng et al. (2020) modify them to RIANTE+ and Li-unified+ for ASTE task. Peng-stage model (Peng et al., 2020) utilizes two sequence tagging schemes to jointly solve ASTE task. JET model (Xu et al., 2020) designs a joint tagging method to identify triplets. GTS model (Wu et al., 2020) utilizes a table filing method to fill a sentiment relation table.

Span-aware methods. Dual-MRC model (Mao et al., 2021) transforms ASTE task into a reading comprehension scheme to extract aspects, opinions, and their corresponding sentiment polarities. BART model (Yan et al., 2021) utilizes a unified framework to decode triplets through a sequence output. PASTE model (Mukherjee et al., 2021) leverages RNN to construct a generative structure, which generates an entire triplet at each time step.

1https://github.com/sentiments-Ananda/MOASTE
Table 4: Comparison F1 score for ASPE, AOPE, and ASTE tasks on $D_{20a}$. The baseline results are retrieved from Li et al. (2019). We highlight the best results in bold.

<table>
<thead>
<tr>
<th>Model</th>
<th>14res</th>
<th>14lap</th>
<th>15res</th>
<th>16res</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ASPE</td>
<td>AOPE</td>
<td>ASTE</td>
<td>ASPE</td>
</tr>
<tr>
<td>Tag</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RINANTE+</td>
<td>48.15</td>
<td>46.29</td>
<td>34.03</td>
<td>41.30</td>
</tr>
<tr>
<td>Li-unified+</td>
<td>73.79</td>
<td>55.34</td>
<td>51.68</td>
<td>65.79</td>
</tr>
<tr>
<td>Peng-stage</td>
<td>74.19</td>
<td>56.10</td>
<td>51.89</td>
<td>62.34</td>
</tr>
<tr>
<td>JET</td>
<td>63.92</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Dual-MRC</td>
<td>76.57</td>
<td>74.93</td>
<td>70.32</td>
<td>64.95</td>
</tr>
<tr>
<td>BART</td>
<td>78.47</td>
<td>77.68</td>
<td>72.46</td>
<td>68.17</td>
</tr>
<tr>
<td>Ours</td>
<td>79.22</td>
<td>78.85</td>
<td>74.53</td>
<td>71.20</td>
</tr>
</tbody>
</table>

Table 5: Comparison P (Precision), R (Recall), F1 score for ASTE task on $D_{20b}$. The baseline results are retrieved from Li et al. (2019), Xu et al. (2021), and Mukherjee et al. (2021).

<table>
<thead>
<tr>
<th>Model</th>
<th>14res</th>
<th>14lap</th>
<th>15res</th>
<th>15res</th>
<th>16res</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tag</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RINANTE+</td>
<td>31.42</td>
<td>39.38</td>
<td>34.95</td>
<td>29.88</td>
<td>30.06</td>
</tr>
<tr>
<td>Li-unified+</td>
<td>41.04</td>
<td>67.35</td>
<td>51.0</td>
<td>44.72</td>
<td>51.39</td>
</tr>
<tr>
<td>Peng-stage</td>
<td>43.24</td>
<td>63.66</td>
<td>51.46</td>
<td>48.07</td>
<td>57.51</td>
</tr>
<tr>
<td>JET</td>
<td>70.56</td>
<td>55.94</td>
<td>62.40</td>
<td>62.59</td>
<td>57.94</td>
</tr>
<tr>
<td>GTS</td>
<td>67.76</td>
<td>67.29</td>
<td>67.50</td>
<td>57.82</td>
<td>51.32</td>
</tr>
<tr>
<td>Span</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BART</td>
<td>65.52</td>
<td>64.99</td>
<td>65.25</td>
<td>61.41</td>
<td>56.19</td>
</tr>
<tr>
<td>Ours</td>
<td>71.78</td>
<td>72.84</td>
<td>72.28</td>
<td>62.83</td>
<td>59.59</td>
</tr>
</tbody>
</table>

4.3 Implementation Details

The uncased English version of BART$_{base}$ is our backbone. We conduct experiments on a single GCU (Nvidia GeForce RTX 2080 Ti) with CUDA version 11.4. The model is trained for 30 epochs with batch size of 8, linear warmup of 1e-1, and weight decay of 1e-2. We use AdamW optimizer with a learning rate of 5e-5. And we use ReLU as the primary activation function in equations. The number of GCN layers is set to 2. We fix the hyperparameters $\alpha$ and $\beta$ as 0.1 and 0.3 for the joint training loss in Equation 17. We report the average results of five runs with different random seeds.

4.4 Main Results

As aforementioned, there are two research lines on ASTE task: tag-aware methods and span-aware methods. For each research line, we compare our method with the above baselines and report results in Table 4 and Table 5.

First, to conduct a detailed evaluation of ASTE (aspect, opinion, sentiment) task, we take AOPE (aspect, opinion) and ASPE (aspect, sentiment) as the special cases of ASTE to verify the effectiveness of our method on $D_{20a}$. Under the F1 metric, we report the results in Table 4 and highlight the best results in bold. Our method dramatically improves these three tasks, that is, 2.06 F1 points for ASPE, 2.07 F1 points for AOPE, and 3.52 F1 points for ASTE on average. The results verify the effectiveness of our method on ASTE task and special cases of it.

Second, we use precision, recall, and F1 score to further evaluate our method for ASTE task on $D_{20b}$. $D_{20b}$ presents a more challenging scenario for overlapped triplets. The results are presented in Table 5. Overall, we still obtain remarkable improvement on four datasets. Our method outperforms the best tag-aware method (i.e., GTS) by an average of 5.13 F1 points. And compared with the best span-aware method (i.e., PASTE), we achieve an average of 4.39 F1 improvement. Besides, we observe that most span-aware methods are superior to tag-aware methods. Tag-aware methods suffer from triplets with multi-word spans because they focus on the interactions between words. Span-aware methods are free from the trouble, but most of them don’t consider the impact of the complex relations between multiple aspects and opinions while decoding three factors of a triplet. Therefore, their performances are worse than our method on ASTE task. In conclusion, all results verify the effectiveness of our method.
5 Analysis & Discussion

5.1 Overlapped Triplets Analysis

We compare our method with PASTE to verify the effectiveness on overlapped triplets. Besides, we evaluate the performance for aspect-overlapped triplets and opinion-overlapped triplets to further identify the area of improvement. Experiment results are shown in Table 6. Our method outperforms PASTE for overlapped triplets and achieves 6.11, 7.5, 7.89, and 5.78 F1 improvements on four datasets. Besides, we obtain an average of 8.82 F1 improvements for aspect-overlapped triplets and an average of 4.39 F1 improvements for opinion-overlapped triplets. The reason for the gap between them is imbalanced data distribution. Besides, we observe that PASTE shows a better recall for opinion-overlapped triplets on 16res, but it is worse than our method due to low precision. This fact suggests that we perform well on overlapped triplets, including both aspect-overlapped triplets and opinion-overlapped triplets. In short, all results show that our method achieves significant improvement for overlapped triplets.

5.2 Multi-Overlap Triplets Analysis

To verify the effectiveness of multi-overlap triplets, we evaluate the performances on $D_{20y}$. The ‘Res’ is a combined dataset from 14res, 15res, and 16res. And the ‘Lap’ comes from 14lap. The results are reported in Table 7. Obviously, our method can effectively solve multi-overlap triplets. We obtain convincing improvement, 6.75 F1 points on the Lap dataset and 6.88 F1 on the Res dataset. Compared to precision, our recall achieves more remarkable improvement than the PASTE. Notably, we surpass PASTE by 8.91 recall points at most on Res dataset. PASTE focuses on the single interactions between an aspect and an opinion when extracting triplets, whereas our method considers the cooperative interactions between multiple aspects and opinions. Therefore, we can better capture more complex relations between aspects and opinions. In conclusion, we gain significant improvement for multi-overlap triplets.

5.3 Joint Decoding Mechanism Efficiency

In the JDM component, we employ the joint decoding of aspects and opinions to promote their generation. To examine if the joint decoding mechanism can effectively generate aspect spans and opinion spans, we evaluate our model for aspect term extraction (ATE) and opinion term extraction (OTE) tasks on $D_{20y}$. As shown in Table 8, we observe that our method outperforms PASTE on four datasets. Moreover, we obtain significant improvements, 3.88 F1 points for ATE task and 3.52 F1 points for OTE task on average. The overall performances verify the effectiveness of the joint decoding mechanism.
Surprisingly, everything is so easy and intuitive to setup or configure.

PASTE: (setup, easy, POS) √, (setup, intuitive, POS) √, (configure, intuitive, POS) √

Ours: (setup, easy, POS) √, (setup, intuitive, POS) √, (configure, intuitive, POS) √

I enjoyed a caesar salad while my wife had (expensive) goat cheese - both very tasty.

PASTE: (caesar salad, enjoy, POS) √, (goat cheese, expensive, POS) ×, (goat cheese, tasty, POS) √

Ours: (caesar salad, enjoy, POS) √, (caesar salad, tasty, POS) √, (goat cheese, expensive, POS) ×, (goat cheese, tasty, POS) √

Table 9: Case study for ASTE task on laptop and restaurant domains. The red and blue indicate the aspects and opinions. Besides, ‘POS’, ‘NEU’, and ‘NEG’ indicate positive, neutral, and negative sentiments.

5.4 Cases Study

We compare our method with PASTE for two cases in Table 9. The first case contains four multi-overlap triplets, and we gain the best performance for them, whereas PASTE misses a triplet. The result indicates that our method can model the complex relations between multiple aspects and opinions. For the second case, our method and PASTE make a mistake on a triplet (goat cheese, expensive, NEG) while predicting its sentiment. The main reason is the imbalanced distribution between positive and negative triplets. Overall, we still outperform PASTE because they miss a multi-overlap triplet again. In short, all cases demonstrate that we can perform well on multi-overlap triplets.

5.5 Ablation Study

We conduct an ablation study to examine the rationality of our method design, and the results are reported in Table 10. The average F1 denotes the results of our method on four datasets over 5 runs. We remove the auxiliary channel, sharing decoder unit, CEN component, and relation-wise calibration. The negative results indicate the absence of any part can decrease our performance. Especially, our performance dropped by 5.20 F1 points on average when the CEN component is replaced with two linear layers. In short, the design of our method is reasonable and achieves the best performance.

<table>
<thead>
<tr>
<th>Model</th>
<th>Average F1</th>
<th>Δ F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full model</td>
<td>67.61</td>
<td></td>
</tr>
<tr>
<td>w/o auxiliary channel</td>
<td>66.30</td>
<td>-1.31</td>
</tr>
<tr>
<td>w/o sharing decoder unit</td>
<td>66.19</td>
<td>-1.42</td>
</tr>
<tr>
<td>w/o CEN component</td>
<td>62.41</td>
<td>-5.20</td>
</tr>
<tr>
<td>w/o relation-wise calibration</td>
<td>64.55</td>
<td>-3.06</td>
</tr>
</tbody>
</table>

Table 10: Comparison of average F1 score for ablation study on D_{20b}.

6 Conclusion

We propose a multi-overlap triplet extraction method to explore the complex relations between multiple aspects and opinions by learning their cooperative interactions. It addresses the limitation that most methods focus on the single interactions between an aspect and an opinion while decoding three factors of a triplet. The ATE and OTE tasks are solved in the decoding process through a joint decoding mechanism. And a correlation-enhanced network reinforces the interactions between related aspects and opinions while predicting their sentiments. Our method obtains convincing improvements on overlapped triplets, especially multi-overlap triplets.

Limitations

Though we obtain convincing performances on multi-overlap triplet extraction, the high time cost is an obvious limitation. The multi-overlap triplet extraction is hugely time-consuming by decoding complex relations under multiple aspects and opinions. Inevitably, our method may be slightly slower than several previous methods. In the follow-up work, we will pay more attention to time consumption.

Acknowledgements

This research is supported by the National Key Research and Development Program of China (No. 2021ZD0111202), the National Natural Science Foundation of China (No.62176005), PKU-Haier "Smart Home" Qingdao Joint Innovation Lab Fund, Project 2020BD002 supported by PKU-Baidu Fund.
References


A Additional Experimental Settings

We utilize PyTorch version 1.7.1 to implement our model and conduct experiments on a single GCU (Nvidia GeForce RTX 2080 Ti) with CUDA version 11.4, NVIDIA-SMI 470.86, and transformer version 3.4.0. The total number of our parameters is 166.72M. During training, the average runtimes (sec/epoch) for ASTE task are shown in Table 11. An aspect sentiment triplet is correct when its three factors match the ground-truth triplet.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>14res</th>
<th>14lap</th>
<th>15res</th>
<th>16res</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D_{20a}$</td>
<td>63</td>
<td>42</td>
<td>29</td>
<td>41</td>
</tr>
<tr>
<td>$D_{20b}$</td>
<td>62</td>
<td>42</td>
<td>28</td>
<td>42</td>
</tr>
</tbody>
</table>

Table 11: Average Runtime