

# Learning Cooperative Interactions for Multi-Overlap Aspect Sentiment Triplet Extraction

ShimanZhao<sup>1,2</sup>, WeiChen<sup>\*1,2,3</sup> and TengjiaoWang<sup>1,2,3</sup>

<sup>1</sup> School of Computer Science, Peking University, Beijing, China

<sup>2</sup> Research Center for Computational Social Science, Peking University

<sup>3</sup> 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:

(*use*, *ease*, *positive*)

(*use*, *never disappoint*, *positive*)

(*service*, *never disappoint*, *positive*)

(*service*, *slightly expensive*, *negative*)

(*service*, *top*, *positive*)

---

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

\*Corresponding author

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.

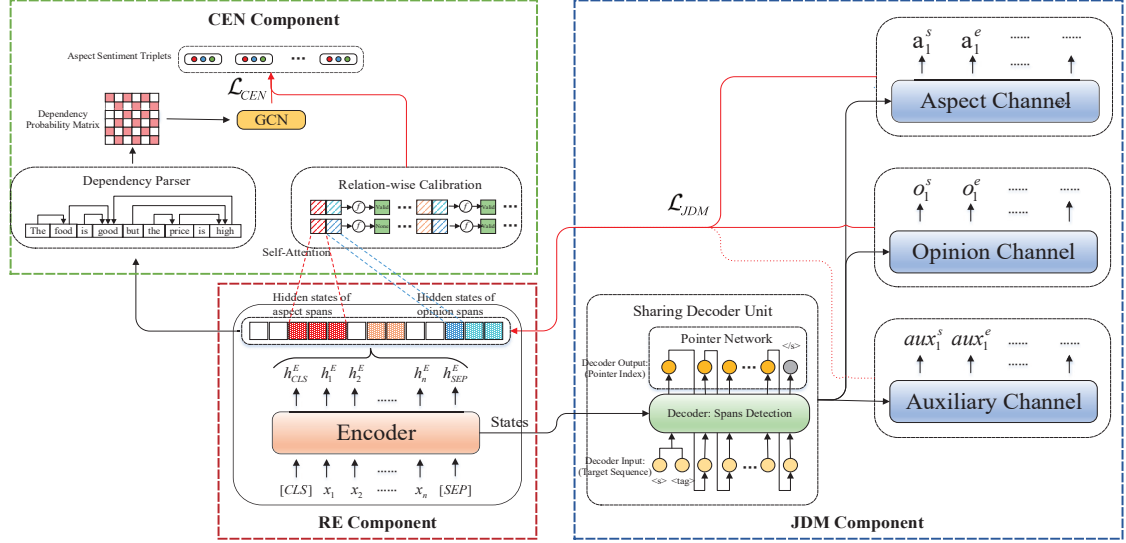


Figure 1: The overall architecture consists of RE, JDM, and CEN components. The top-left green dotted box refers to the CEN component, the bottom-left red dotted box refers to the RE component, and the right blue dotted box refers to the JDM component.

### 3 Method

#### 3.1 Task Formulation

For ASTE task, given a sentence  $X = \{x_1, x_2, \dots, x_n\}$ . The sentence  $X$  is a sequence of words,  $x_i$  is the  $i^{\text{th}}$  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  $T^i = (a_i^s, a_i^e, o_i^s, o_i^e, s_i^p)$ . 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, which constructs an input sentence 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.

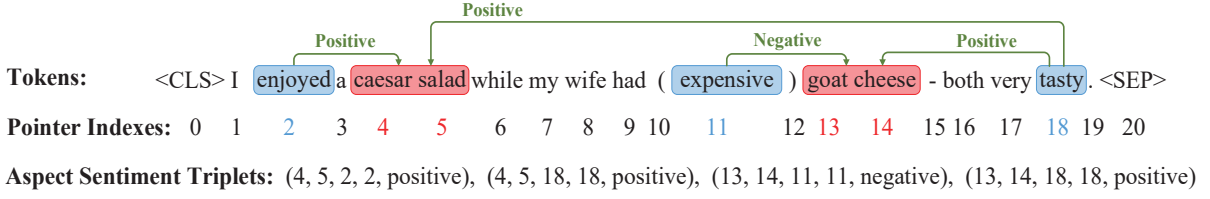
**CEN.** The CEN component consists of an interaction enhancement module and a relation-wise calibration scheme. The interaction enhancement module encodes the context with dependency relations to reinforce the interactions between related aspects and opinions while predicting their sentiments. And the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

##### 3.2.1 The RE component

For a sentence  $X = \{x_1, x_2, \dots, x_n\}$ , 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  $H^E = [h_{CLS}^E, h_X^E, h_{SEP}^E]$ , where  $h_X^E = [h_{x_1}^E, h_{x_2}^E, \dots, h_{x_n}^E]$ . The  $H^E$  is as follows:

$$H^E = RE([CLS], x_1, x_2, \dots, x_n, [SEP]) \quad (1)$$

where  $H^E \in R^{n*d}$ , and  $d$  is the hidden dimension.



Span Generation Channels	Target Sequences
Aspect Channel	[s], [AC], 4, 5, 13, 14, [/s]
Opinion Channel	[s], [OC], 2, 2, 11, 11, 18, 18, [/s]
Auxiliary Channel	[s], [AuxC], 0, 1, 3, 3, 6, 10, 12, 12, 15, 17, 19, 20, [/s]

Figure 2: The target sequence examples for the aspect channel, the opinion channel, and the auxiliary channel. During training, we convert channel symbol tokens (i.e., [AC], [OC], and [AuxC]) and the special start token and end token (i.e., [s] and [/s]) to corresponding class indexes.

### 3.2.2 The JDM component

We divide the whole sentence into aspect spans, opinion spans, and other spans. Besides, we use ( $aux^s$ ,  $aux^e$ ) to represent other spans. Then, we construct target sequences for three channels.

To identify different channels, we add channel symbol tokens [AC], [OC], and [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:**  
 $Y = [s], [AC], a_1^s, a_1^e, \dots, a_i^s, a_i^e, [/s]$
- **Opinion Channel:**  
 $Y = [s], [OC], o_1^s, o_1^e, \dots, o_j^s, o_j^e, [/s]$
- **Auxiliary Channel:**  
 $Y = [s], [AuxC], aux_1^s, aux_1^e, \dots, aux_k^s, aux_k^e, [/s]$

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  $h_t^D$ .

$$h_t^D = \text{Decoder}(H^E, \hat{Y}_{<t}) \quad (2)$$

where  $h_t^D \in R^d$ , the probability distribution  $P_t$  is as follows.

$$H^{E'} = \text{MLP}(H^E) \quad (3)$$

$$M^D = \text{Decoder.embed\_tokens}(X) \quad (4)$$

$$M^{D'} = \text{Dropout}(M^D) \quad (5)$$

$$P_t = \text{Softmax}\left(\frac{1}{2}(H^{E'} + M^{D'})h_t^D\right) \quad (6)$$

where  $H^E, H^{E'}, M^D, M^{D'} \in R^{n \times d}$ ,  $h_t^D \in R^{1 \times d}$ , 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  $Y^a = \{y_1^a, y_2^a, \dots, y_{T_a}^a\}$ , the target sequence of the opinion channel as  $Y^o = \{y_1^o, y_2^o, \dots, y_{T_o}^o\}$ , and the target sequence of the auxiliary channel as  $Y^{aux} = \{y_1^{aux}, y_2^{aux}, \dots, y_{T_{aux}}^{aux}\}$ . Then, we utilize a cross-entropy for optimization with  $\mathcal{L}_{JDM}$ :

$$\mathcal{L}_{JDM} = - \sum_{t=1}^{T_a} (y_t^a \log(\hat{y}_t^a) + \sum_{t=1}^{T_o} y_t^o \log(\hat{y}_t^o) + \sum_{t=1}^{T_{aux}} y_t^{aux} \log(\hat{y}_t^{aux})) \quad (7)$$

During inference, we put the start token [s] and the channel symbol token (i.e., [AC], [OC], or [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 auto-regressive 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		$D_{20_a}$								$D_{20_b}$							
		#S	#MulPol	#OverLap	triplets in a sentence					#S	#MulPol	#OverLap	triplets in a sentence				
					1	2	3	4	$\geq 5$				1	2	3	4	$\geq 5$
14res	train	1300	107	264	715	393	141	36	15	1266	122	388	605	397	165	67	32
	dev	323	26	59	181	102	24	13	3	310	27	106	153	91	39	18	9
	test	496	28	122	245	173	50	20	8	492	35	179	206	172	60	30	24
14lap	train	920	37	147	653	203	54	7	3	906	47	269	545	228	93	27	13
	dev	228	5	53	146	61	17	2	2	219	10	63	133	62	14	5	5
	test	339	14	62	226	83	22	8	0	328	18	99	184	101	23	15	5
15res	train	593	27	123	358	167	43	23	2	605	31	165	338	172	58	31	6
	dev	148	11	33	84	44	16	3	1	148	8	45	76	51	15	5	1
	test	318	24	48	221	66	23	7	1	322	25	68	210	77	22	10	3
16res	train	842	52	164	528	213	72	26	3	857	52	220	504	230	73	41	9
	dev	210	9	39	133	58	11	7	1	210	10	56	118	63	23	5	1
	test	320	10	58	204	93	18	4	1	326	11	79	192	97	23	11	3

Table 2: The sentence-level statistics of four datasets on  $D_{20_a}$  and  $D_{20_b}$ . #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  $\geq 5$ ) denote the number of sentences with 1, 2, 3, 4, and more triples.

Datasets		$D_{20_a}$						$D_{20_b}$							
		#S	#T	+	0	-	#AOT	#OOT	#S	#T	+	0	-	#AOT	#OOT
14res	train	1300	2145	1575	143	427	568	33	1266	2337	1506	148	453	542	482
	dev	323	524	377	32	115	114	16	310	577	352	48	111	144	145
	test	496	862	675	45	142	258	16	492	994	666	50	146	274	242
14lap	train	920	1265	664	117	484	300	18	906	1460	696	111	480	355	362
	dev	228	337	207	16	114	112	6	219	345	149	31	124	98	82
	test	339	490	335	50	105	131	7	328	541	316	54	110	147	129
15res	train	593	923	703	25	195	258	20	605	1013	722	25	195	279	144
	dev	148	238	179	9	50	64	6	148	249	174	9	53	69	27
	test	318	455	291	25	139	98	8	322	485	297	25	139	100	55
16res	train	842	1289	933	49	307	336	30	857	1394	942	48	317	365	178
	dev	210	316	225	10	81	82	4	210	339	235	11	73	81	45
	test	320	465	362	27	76	114	10	326	514	371	28	76	118	71

Table 3: The triplet-level statistics of four datasets on  $D_{20_a}$  and  $D_{20_b}$ , 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 from the parser tree (Mrini et al., 2019). Then, we leverage the GCN to encode the context with dependency relations. The equation (Li et al., 2021) is as follows.

$$h_i^l = \sigma \left( \sum_{j=1}^n A_{ij} W^l h_j^{l-1} + b^l \right) \quad (8)$$

where  $A$  is a dependency probability matrix, and  $h_i^l$  is the  $i^{th}$  node at the  $l^{th}$  layer. And the initial representation of  $h_i^l$  comes from  $H^E$ .  $W^l$  and  $b^l$  are learnable parameters.  $\sigma$  is an activation function.

We get enhanced states  $H^G = [h_1^G, h_2^G, \dots]$ . Then, we concatenate the start word representations of the aspect and opinion spans to predict

their sentiments. The equations are as follows.

$$sr_{ao}^G = [h_{a^s}^G; h_{o^s}^G] \quad (9)$$

$$p(m|sr_{ao}^G) = Soft \max((f(W_m^1 sr_{ao}^G + b_m^1) W_m^2 + b_m^2)) \quad (10)$$

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  $m \in \{Positive, Negative, Neutral, None\}$ .

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:

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

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

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

where  $M_{sr} \in R^{L*d}$  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  $SR_{ao} = \{sr_{ao}^1, sr_{ao}^2, \dots, sr_{ao}^q\}$ , where  $sr_{ao}^q$  denotes the  $q^{th}$  aspect-opinion pair representation:

$$sr_{ao}^q = [sr_a^i; sr_o^j; sr_a^i - sr_o^j] \quad (14)$$

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.

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

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

We utilize log-likelihood to optimize  $\mathcal{L}_{CEN}$ :

$$\begin{aligned} \mathcal{L}_{CEN} = & - \sum ( \sum_{sr_{ao} \in SR_{ao}} \log P(r^* | sr_{ao}) \\ & + \sum_{sr_{ao}^G \in SR_{ao}^G} \log P(m^* | sr_{ao}^G) ) \end{aligned} \quad (16)$$

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.

$$\mathcal{L} = \alpha \mathcal{L}_{JDM} + \beta \mathcal{L}_{CEN} \quad (17)$$

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.  $D_{20b}$  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 <sup>1</sup>.

### 4.2 Baselines

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

**Tag-aware methods.** RINANTE 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 RINANTE+ 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.

<sup>1</sup><https://github.com/sentiments-Ananda/MOASTE>

Model	14res			14lap			15res			16res			
	ASPE	AOPE	ASTE	ASPE	AOPE	ASTE	ASPE	AOPE	ASTE	ASPE	AOPE	ASTE	
Tag	RINANTE+	48.15	46.29	34.03	36.70	29.70	20.0	41.30	35.40	28.0	42.10	30.70	
	Li-unified+	73.79	55.34	51.68	63.38	52.56	42.47	64.95	56.85	46.69	70.20	53.75	44.51
	Peng-stage	74.19	56.10	51.89	62.34	53.85	43.50	65.79	56.23	46.79	71.73	60.04	53.62
	JET	-	-	63.92	-	-	50.0	-	-	54.67	-	-	62.98
Span	Dual-MRC	76.57	74.93	70.32	64.59	63.37	55.58	65.14	64.97	57.21	70.84	75.71	67.40
	BART	78.47	77.68	72.46	68.17	66.11	57.59	69.95	67.98	60.11	75.69	77.38	69.98
	Ours	<b>79.22</b>	<b>78.85</b>	<b>74.53</b>	<b>71.20</b>	<b>69.94</b>	<b>62.30</b>	<b>71.79</b>	<b>69.69</b>	<b>63.10</b>	<b>78.29</b>	<b>78.94</b>	<b>74.27</b>

Table 4: Comparison F1 score for ASPE, AOPE, and ASTE tasks on  $D_{20_a}$ . The baseline results are retrieved from Li et al. (2019). We highlight the best results in bold.

Model	14res			14lap			15res			16res			
	P.	R.	F1.	P.	R.	F1.	P.	R.	F1.	P.	R.	F1.	
Tag	RINANTE+	31.42	39.38	34.95	21.71	18.66	20.07	29.88	30.06	29.97	25.68	22.3	23.87
	Li-unified+	41.04	67.35	51.0	40.56	44.28	42.34	44.72	51.39	47.82	37.33	54.51	44.31
	Peng-stage	43.24	63.66	51.46	37.38	50.38	42.87	48.07	57.51	52.32	46.96	64.24	54.21
	JET	70.56	55.94	62.40	55.39	47.33	51.04	<b>64.45</b>	51.96	57.53	70.42	58.37	63.83
	GTS	67.76	67.29	67.50	57.82	51.32	54.36	62.59	57.94	60.15	66.08	69.91	67.93
Span	PASTE	68.70	63.80	66.10	59.70	55.30	57.40	63.60	59.80	61.60	68.00	67.70	67.80
	BART	65.52	64.99	65.25	61.41	56.19	58.69	59.14	59.38	59.26	66.6	68.68	67.62
	Ours	<b>71.78</b>	<b>72.84</b>	<b>72.28</b>	<b>62.83</b>	<b>59.59</b>	<b>61.16</b>	63.53	<b>62.77</b>	<b>63.06</b>	<b>72.73</b>	<b>75.23</b>	<b>73.95</b>

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

### 4.3 Implementation Details

The uncased English version of BART<sub>base</sub> 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. And we use AdamW optimizer with a learning rate of 5e-5. The dropout rate is 0.5 in Equation 5. And ReLU is 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  $D_{20_a}$ . 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  $D_{20_b}$ .  $D_{20_b}$  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.

Type	Model	14res			14lap			15res			16res		
		P.	R.	F.	P.	R.	F.	P.	R.	F.	P.	R.	F.
OverlapTriplet	PASTE	83.40	60.76	70.34	78.58	45.52	57.56	72.64	49.06	58.50	80.80	64.14	71.54
	Ours	85.05	69.43	<b>76.45</b>	79.74	54.96	<b>65.06</b>	77.55	58.04	<b>66.39</b>	87.06	69.54	<b>77.32</b>
AspectOverlap	PASTE	86.38	62.04	72.32	83.42	44.12	57.18	68.84	50.00	57.88	81.80	62.66	70.82
	Ours	88.64	71.53	<b>79.16</b>	85.37	53.52	<b>65.78</b>	79.07	64.40	<b>70.97</b>	86.96	70.00	<b>77.55</b>
OpinionOverlap	PASTE	80.23	52.64	63.54	72.32	46.30	55.56	69.72	41.84	52.36	80.74	64.84	71.96
	Ours	80.94	61.23	<b>69.71</b>	75.66	52.27	<b>61.82</b>	72.10	45.11	<b>55.48</b>	87.29	64.19	<b>73.98</b>

Table 6: Comparison P (Precision), R (Recall), F1 score for overlapped triplets (i.e., ‘OverlapTriplet’), aspect-overlapped triplets (i.e., ‘AspectOverlap’) and opinion-overlapped triplets (i.e., ‘OpinionOverlap’) on  $D_{20_b}$ .

Model	Lap			Res		
	P.	R.	F.	P.	R.	F.
PASTE	72.74	43.30	54.20	70.52	41.90	52.50
Ours	77.56	50.27	60.95	71.69	50.81	59.38
$\Delta$	4.82	6.97	6.75	1.17	8.91	6.88

Table 7: Comparison P (Precision), R (Recall), F1 score for sentences with multi-overlap triplets on datasets Res and Lap.  $\Delta$  indicates the gap value between ours and PASTE. We report the average results of 5 runs.

## 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  $D_{20_b}$ . 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

Dataset	Model	ATE			OTE		
		P.	R.	F.	P.	R.	F.
14res	PASTE	81.94	81.24	81.94	81.96	81.26	81.60
	Ours	84.09	85.99	<b>85.03</b>	83.76	87.12	<b>85.40</b>
14lap	PASTE	83.54	79.56	80.92	78.74	71.84	75.12
	Ours	84.96	84.41	<b>84.67</b>	80.59	78.98	<b>79.76</b>
15res	PASTE	81.74	76.46	78.92	78.56	73.92	76.42
	Ours	84.04	82.58	<b>83.30</b>	80.04	80.04	<b>79.97</b>
16res	PASTE	78.72	85.10	81.72	82.18	85.76	83.94
	Ours	83.41	88.79	<b>86.01</b>	83.67	88.56	<b>86.04</b>

Table 8: Comparison P (Precision), R (Recall), F1 score for ATE and OTE tasks on  $D_{20_b}$ .

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  $D_{20_b}$ . 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.



<b>Sentence1:</b>	Surprisingly, everything is so <b>easy</b> and <b>intuitive</b> to <b>setup</b> or <b>configure</b> .
<b>PASTE:</b>	(setup, easy, POS) ✓, (setup, intuitive, POS) ✓, (configure, intuitive, POS) ✓
<b>Ours:</b>	(setup, easy, POS) ✓, (setup, intuitive, POS) ✓, (configure, intuitive, POS) ✓ (configure, easy, POS) ✓
<b>Sentence2:</b>	I <b>enjoyed</b> a <b>caesar salad</b> while my wife had ( <b>expensive</b> ) <b>goat cheese</b> - both very <b>tasty</b> .
<b>PASTE:</b>	(caesar salad, enjoy, POS) ✓, (goat cheese, expensive, POS) ×, (goat cheese, tasty, POS) ✓
<b>Ours:</b>	(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.

Model	Average F1	$\Delta$ F1
Full model	<b>67.61</b>	
w/o auxiliary channel	66.30	-1.31
w/o sharing decoder unit	66.19	-1.42
w/o CEN component	62.41	-5.20
w/o relation-wise calibration	64.55	-3.06

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

- John Atkinson-Abutridy. 2022. *Natural-Language Processing*, pages 15–48.
- Shaowei Chen, Yu Wang, Jie Liu, and Yuelin Wang. 2021a. Bidirectional machine reading comprehension for aspect sentiment triplet extraction. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, pages 12666–12674.
- Zhexue Chen, Hong Huang, Bang Liu, Xuanhua Shi, and Hai Jin. 2021b. Semantic and syntactic enhanced aspect sentiment triplet extraction. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1474–1483.
- Hongliang Dai and Yangqiu Song. 2019. Neural aspect and opinion term extraction with mined rules as weak supervision. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5268–5277.
- Lei Gao, Yulong Wang, Tongcun Liu, Jingyu Wang, Lei Zhang, and Jianxin Liao. 2021. Question-driven span labeling model for aspect–opinion pair extraction. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, pages 12875–12883.
- Mengting Hu, Shiwan Zhao, Honglei Guo, Chao Xue, Hang Gao, Tiegang Gao, Renhong Cheng, and Zhong Su. 2021. Multi-label few-shot learning for aspect category detection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 6330–6340.
- Ruifan Li, Hao Chen, Fangxiang Feng, Zhanyu Ma, Xiaojie Wang, and Eduard Hovy. 2021. Dual graph convolutional networks for aspect-based sentiment analysis. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, pages 6319–6329.
- Xin Li, Lidong Bing, Piji Li, and Wai Lam. 2019. A unified model for opinion target extraction and target sentiment prediction. In *Proceedings of the 33th AAAI conference on artificial intelligence*, pages 6714–6721.
- Yunlong Liang, Fandong Meng, Jinchao Zhang, Yufeng Chen, Jinan Xu, and Jie Zhou. 2020. An iterative multi-knowledge transfer network for aspect-based sentiment analysis. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1768–1780.
- Jian Liu, Zhiyang Teng, Leyang Cui, Hanmeng Liu, and Yue Zhang. 2021. Solving aspect category sentiment analysis as a text generation task. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4406–4416.
- Yue Mao, Yi Shen, Chao Yu, and Longjun Cai. 2021. A joint training dual-mrc framework for aspect based sentiment analysis. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*.
- K. Mrini, F. Derroncourt, T. Bui, W. Chang, and N. Nakashole. 2019. Rethinking self-attention: An interpretable self-attentive encoder-decoder parser. *CoRR*, abs/1911.03875.
- Rajdeep Mukherjee, Tapas Nayak, Yash Butala, Sourangshu Bhattacharya, and Pawan Goyal. 2021. Paste: A tagging-free decoding framework using pointer networks for aspect sentiment triplet extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9279–9291.
- Haiyun Peng, Lu Xu, Lidong Bing, Fei Huang, Wei Lu, and Luo Si. 2020. Knowing what, how and why: A near complete solution for aspect-based sentiment analysis. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, pages 8600–8607.
- Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphee de clercq, Véronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeny Kotelnikov, Nuria Bel, Salud María Zafra, and Gülşen Eryiğit. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. pages 19–30.
- Hai Wan, Yufei Yang, J.F. Du, Yanan Liu, Kunxun Qi, and Jeff Pan. 2020. Target-aspect-sentiment joint detection for aspect-based sentiment analysis. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:9122–9129.
- Qianlong Wang, Zhiyuan Wen, Qin Zhao, Min Yang, and Ruifeng Xu. 2021. Progressive self-training with discriminator for aspect term extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 257–268.
- Zhen Wu, Chengcan Ying, Fei Zhao, Zhifang Fan, Xinyu Dai, and Rui Xia. 2020. Grid tagging scheme for aspect-oriented fine-grained opinion extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2576–2585.
- H. Xu, B. Liu, L. Shu, and Philip S Yu. 2018. Double embeddings and cnn-based sequence labeling for aspect extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 592–598.
- Lu Xu, Yew Ken Chia, and Lidong Bing. 2021. Learning span-level interactions for aspect sentiment triplet extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, pages 4755–4766.
- Lu Xu, Hao Li, Wei Lu, and Lidong Bing. 2020. Position-aware tagging for aspect sentiment triplet extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 2339–2349.

- Hang Yan, Junqi Dai, Xipeng Qiu, Zheng Zhang, et al. 2021. A unified generative framework for aspect-based sentiment analysis. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, pages 2416–2429.
- Jianfei Yu, Jing Jiang, and Rui Xia. 2018. Global inference for aspect and opinion terms co-extraction based on multi-task neural networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, PP:1–1.
- Chen Zhang, Qiuchi Li, Dawei Song, and Benyou Wang. 2020a. A multi-task learning framework for opinion triplet extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 819–828.
- Ji Zhang, Chengyao Chen, Pengfei Liu, Chao He, and Cane Wing-Ki Leung. 2020b. Target-guided structured attention network for target-dependent sentiment analysis. In *Transactions of the Association for Computational Linguistics*, pages 172–182.
- Wenxuan Zhang, Yang Deng, Xin Li, Yifei Yuan, Lidong Bing, and Wai Lam. 2021. Aspect sentiment quad prediction as paraphrase generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9209–9219.

## 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.

<b>Dataset</b>	<b>14res</b>	<b>14lap</b>	<b>15res</b>	<b>16res</b>
$D_{20_a}$	63	42	29	41
$D_{20_b}$	62	42	28	42

Table 11: Average Runtime