On the Strength of Sequence Labeling and Generative Models for Aspect Sentiment Triplet Extraction

Shen Zhou[1], **Tieyun Qian**[1,2,*]
[1]School of Computer Science, Wuhan University, China
[2]Intellectual Computing Laboratory for Cultural Heritage, Wuhan University, China
{shenzhou, qty}@whu.edu.cn

Abstract

Generative models have achieved great success in aspect sentiment triplet extraction tasks. However, existing methods ignore the mutual informative clues between aspect and opinion terms and may generate false paired triplets. Furthermore, the inherent limitations of generative models, i.e., the token-by-token decoding and the simple structured prompt, prevent models from handling complex structures especially multi-word terms and multi-triplet sentences. To address these issues, we propose a sequence labeling enhanced generative model. Firstly, we *encode the dependency between aspect and opinion into two bidirectional templates* to avoid false paired triplets. Secondly, we *introduce a marker-oriented sequence labeling module* to improve generative models' ability of tackling complex structures. Specifically, this module enables the generative model to capture the boundary information of aspect/opinion spans and provides hints to decode multiple triplets with the shared marker. Experimental results on four datasets prove that our model yields a new state-of-art performance. Our code and data are available at https://github.com/NLPWM-WHU/SLGM.

1 Introduction

Aspect sentiment triplet extraction (ASTE) aims at extracting all triplets in a sentence, consisting of the aspect/opinion terms and the sentiment polarity on them. Given the example "*Their twist on pizza is healthy, but full of flavor.*" in Fig. 1 (a), the goal is to extract two triplets (twist on pizza, healthy, positive) and (flavor, full, positive).

Conventional approaches to ASTE include pipeline (Peng et al., 2020), table filling (Chen et al., 2022), sequence tagging (Xu et al., 2020; Wu et al., 2020b), and hybrid ones (Xu et al., 2021). More recently, there is an emerging trend in adopting generative models for ASTE (Yan et al., 2021; Zhang et al., 2021b,a; Lu et al., 2022) to alleviate error propagation and exploit full label semantics.

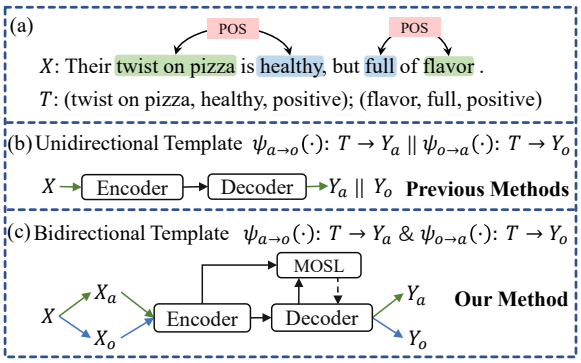

Figure 1: (a) shows an example for the ASTE task. (b) and (c) illustrate the difference between our proposed generative method and existing ones for this task, where X is the input sentence, and T denotes the target triplet. X_a/X_o contains the prompt prefix to define the decoding order (*a*spect or *o*pinion first) while Y_a/Y_o indicates the generated sequences following the order in X_a/X_o. MOSL is our marker-oriented sequence labeling module to improve the generative model's ability of handling complex structures.

Current generative ASTE models employ a classical encoder-decoder architecture and follow a paradigm that first generates a target sequence Y and then recovers the triplets T from the sequence Y. The model needs to pre-define an output template $\psi(\cdot)$ to convert ASTE into text generation and then calculates the loss between the triplet and the generated sequence for model training, as shown in Fig. 1 (b). The template $\psi(\cdot)$ constructed by existing methods is in the form of $\psi_{a\to o}$ or $\psi_{o\to a}$, reflecting the unidirectional dependency from aspect to opinion, or vice versa. However, the aspect and opinion terms that appear together in one sentence might hold informative clues to each other (Chen and Qian, 2020b) and there is no intrinsic order between them (Chen et al., 2021). Hence, modeling unidirectional dependency may mislead the model to generate false paired triplets like (*twist on pizza*, *full*, *positive*).

* Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2023, pages 12038–12050

Existing generative ASTE models also suffer from another challenging problem, i.e., lacking the ability to handle complex structures especially multi-word terms and multi-triplet sentences. On one hand, the token-by-token decoding manner makes the model focus only on the next token at each time step of decoding without grasping the whole information of the aspect/opinion term with multiple words. On the other hand, generative models often deploy the simple-structured prompt template to ensure the generation quality. When handling the sentence with multiple triplets, a generative model needs to invoke a template several times, which may lead to an information confusion for the same marker in the template.

To address the aforementioned issues, we propose a sequence labeling enhanced generative model for ASTE.

Firstly, we design two bidirectional templates with different decoding orders to simultaneously capture the mutual dependency between the aspect and opinion terms. In particular, we add two types of prompt prefix before the input sentence to indicate the decoding order, and we also present two output templates $\psi_{a\to o}$ and $\psi_{o\to a}$, both consisting of the markers {aspect, opinion, sentiment} and the corresponding labels {a, o, s}. In this way, the decoder can generate two sentences reflecting dependency from aspect to opinion and that from opinion to aspect.

Secondly, we propose a marker-oriented sequence labeling (MOSL) module, which can enhance the generative model's ability to handle complex structures. Specifically, the decoding is conducted after the MOSL module at the training stage. Hence the BIO tags obtained in MOSL help the generative model capture the boundary information of multi-word aspect/opinion terms in advance. Moreover, while the generative model needs to invoke the output templates several times for the multi-triplet sentence, we adopt different marker vectors in MOSL for the same marker in the generative model. By doing this, we can share the markers without causing confusion. Since the markers encode information across multiple triplets in one sentence, previous markers can contribute to the decoding of subsequent triplets. The illustration of our proposed method is shown in Fig. 1 (c).

We conduct extensive experiments on four datasets with both full supervised and low-resource settings. The results demonstrate that our model significantly outperforms the state-of-art baselines for the ASTE task.

2 Related Work

Aspect-based sentiment analysis traditionally involves three basic tasks, including aspect extraction (Xu et al., 2018; Dai and Song, 2019; Chen and Qian, 2020a), aspect-level sentiment classification (Zhang and Qian, 2020; Zhou et al., 2021; Li et al., 2021), and opinion extraction (Wu et al., 2020a).

To meet the practical need, some recent studies propose to extract two or more elements simultaneously, including aspect opinion pair extraction (Zhao et al., 2020; Wu et al., 2021; Gao et al., 2021), end-to-end aspect-based sentiment analysis (Hu et al., 2019; Chen and Qian, 2020b; Oh et al., 2021), and aspect sentiment triplet extraction. Among them, ATSE is regarded as a near complete task and is of the most challenge.

Earlier work in ATSE can be sorted into four streams, i.e., pipeline (Peng et al., 2020), table filling (Chen et al., 2022), sequence tagging (Xu et al., 2020; Wu et al., 2020b), and hybrid ones (Xu et al., 2021; Chen et al., 2021; Mao et al., 2021). These methods do not fully utilize the rich label semantics and some of them may encounter the error propagation problem.

Another line of research in ASTE performs this task in a generative manner (Zhang et al., 2021a,b). For example, Yan et al. (2021) model the extraction and classification tasks as the generation of pointer indexes and class indexes. Lu et al. (2022) introduce the structured extraction language and structural schema instructor to unify all information extraction tasks. While getting better performance, current generative models are prone to generate false paired triplets and are not suitable for tackling complex structures. Our generative model addresses these issues with the proposed bidirectional templates and the marker-oriented sequence labeling module.

3 Our Method

Given a review sentence X with L words, the goal of ASTE is to extract all triplets $T = \{(a, o, s)\}_{i=1}^{N}$ in X, where N is the number of triplets, and a, o, and s denotes aspect term, opinion term, and sentiment polarity, respectively.

We first introduce the overall architecture of our proposed sequence labeling enhanced generative

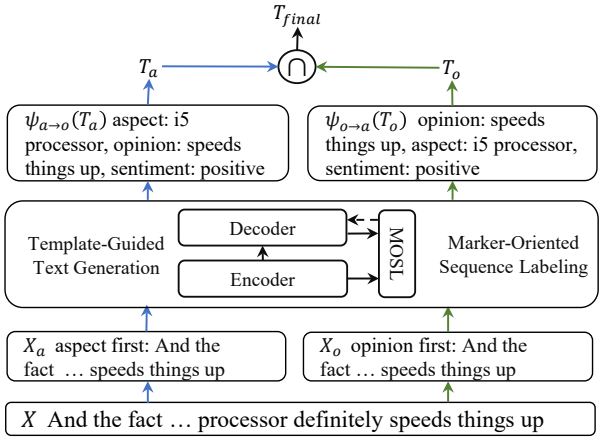

Figure 2: The overall architecture of our sequence labeling enhanced generative model (SLGM).

model (SLGM) in Fig. 2, which has the following distinguished characteristics.

(1) To capture the mutual information between the aspect and opinion terms, we construct two bidirectional templates at both the input and output ends, shown as X_a/X_o and $\psi_{a\to o}$/$\psi_{o\to a}$ in Fig. 2.

(2) To handle complex structures, we propose a marker-oriented sequence labeling (MOSL) module to capture the boundary information of multi-word aspect/opinion terms and the shared marker information of multi-triplets.

3.1 Bidirectional Template

Our bidirectional templates are used to guide the generation model in an end-to-end way.

For the input review X, we construct two sentences X_a and X_o by adding two types of prompt prefix, i.e, "aspect first:" and "opinion first:". Such prefix can prompt the model to generate target sequence with specific decoding order when we fine-tune the model with these templates.

To get the output triplets T in a generative manner, an essential step is linearizing triplets T into a target sequence during training and de-linearizing triplets from the predicted sequence during inference. In particular, a good output template is expected to: 1) ensure that the linearized target sequence can be easily de-linearized into a collection of triples, 2) contain specific markers to prompt the decoding process of labels, 3) free to change the order of labels. Based on the above considerations, we propose two marker-based templates $\psi_{a\to o}$ and $\psi_{o\to a}$ with different decoding orders between aspect and opinion terms as follows:

$$\psi_{a\to o} \to \textbf{aspect} : a, \textbf{opinion} : o, \textbf{sentiment} : s$$
$$\psi_{o\to a} \to \textbf{opinion} : o, \textbf{aspect} : a, \textbf{sentiment} : s$$

Our output templates consist of two parts: the markers {aspect, opinion, sentiment} and the corresponding labels $\{a, o, s\}$. The markers can guide the model to generate the specific type of label at the next step. When the input review contains several triplets, we need to sort the triplet order to ensure the uniqueness of the target sequence. For the template $\psi_{a\to o}$, we sort triplets by the end index of aspect term in an ascending order. If some triplets share the same aspect term, we further sort them by the end index of opinion term. After obtaining text segments of triplets, we use a special symbol `[SSEP]` to concatenate these segments to form the final target sequence.

3.2 Template-Guided Text Generation

We employ a standard transformer-based encoder-decoder architecture for the text generation process, and we initialize the model's parameters with the pre-trained language model T5 (Raffel et al., 2020). For simplicity, we take the sentence X_a and the corresponding target sequence Y_a based on the template $\psi_{a\to o}$ as an example for illustration. We first feed X_a into the transformer encoder to get contextual features $\mathbf{H^{enc}}$:

$$\mathbf{H^{enc}} = \mathbf{Encoder}(X_a) \tag{1}$$

We then use a transformer decoder to generate the target sequence Y_a. At the t-th time step, the decoder will calculate the decoder hidden states $\mathbf{h}_t$ based on the contextual features $\mathbf{H^{enc}}$ and the previously decoded tokens $y_{[1:t-1]}$.

$$\mathbf{h}_t = \mathbf{Decoder}(y_{[1:t-1]}, \mathbf{H^{enc}}) \tag{2}$$

Next, $\mathbf{h}_t$ is used to compute the conditional probability of the token y_t:

$$p(y_t|\mathbf{H^{enc}}; y_{[1:t-1]}) = softmax(\mathbf{W^T h}_t), \tag{3}$$

where $\mathbf{W}$ is the transformation matrix. Finally, we calculate the cross-entropy loss $\mathcal{L}_g^{a\to o}$ between the decoder output and the target sequence Y_a:

$$\mathcal{L}_g^{a\to o} = -\sum_{i=1}^{L} \log p(y_t|\mathbf{H^{enc}}; y_{[1:t-1]}) \tag{4}$$

3.3 Marker-Oriented Sequence Labeling (MOSL)

The marker-based templates can prompt the generative model with the label types including aspect, opinion, and sentiment. However, the classic encoder-decoder architecture prevents the model from handling complex structures. On one hand, the decoding process is performed in a token-by-token manner, which cannot provide clear bound-

ary information for multi-word aspect/opinion terms. On the other hand, the model needs to invoke the output templates repeatedly when the sentence contains multiple triplets. The duplicate template based decoding may cause an information confusion and sacrifice the quality of the generated text. Therefore, we propose a marker-oriented sequence labeling (MOSL) module to solve these problems. The goal is to allow the model to incorporate the prompt information of aspect and opinion terms during the generation of the specific marker [1]. Fig. 3 illustrates the text generation process enhanced by the marker-oriented sequence labeling (MOSL) module.

In MOSL, we will tag aspect and opinion terms through sequence labeling. We first use two linear transformations to extract aspect features $\mathbf{H}^a = \{\mathbf{h}_1^a, \mathbf{h}_2^a, \cdots, \mathbf{h}_L^a\} \in \mathbb{R}^{L\times d}$ (L is the sentence length) and opinion features $\mathbf{H}^o = \{\mathbf{h}_1^o, \mathbf{h}_2^o, \cdots, \mathbf{h}_L^o\} \in \mathbb{R}^{L\times d}$ from the contextual features $\mathbf{H}^{\mathbf{enc}}$:

$$\mathbf{H}^a = \mathrm{MLP}_a(\mathbf{H}^{\mathbf{enc}}),\ \mathbf{H}^o = \mathrm{MLP}_o(\mathbf{H}^{\mathbf{enc}}) \quad (5)$$

Then, we take the last hidden state of the decoder corresponding to the markers as the marker features, including aspect marker features $\mathcal{M}^a = \{\mathbf{m}_1^a, \mathbf{m}_2^a, \cdots, \mathbf{m}_N^a\}$ (N is the number of triplets) and opinion marker features $\mathcal{M}^o = \{\mathbf{m}_1^o, \mathbf{m}_2^o, \cdots, \mathbf{m}_N^o\}$. We then calculate the marker-oriented features for $\mathbf{m}_i^a \in \mathcal{M}^a$ or $\mathbf{m}_i^o \in \mathcal{M}^o$ for sequence labeling:

$$\begin{aligned}\mathbf{q}_{ij}^a &= \sigma(\mathbf{W}_1(\mathbf{h}_j^a \oplus \mathbf{m}_i^a) + \mathbf{b}_1),\\ \mathbf{q}_{ij}^o &= \sigma(\mathbf{W}_1(\mathbf{h}_j^o \oplus \mathbf{m}_i^o) + \mathbf{b}_1),\end{aligned} \quad (6)$$

where $\sigma(\cdot)$ is the selu activation function, $\mathbf{h}_j^a \in \mathbf{H}^a$ and $\mathbf{h}_j^o \in \mathbf{H}^o$ are the aspect/opinion features[2]. $\mathbf{W}$ and $\mathbf{b}$ are the transformation matrix and bias.

Note that we deploy a tag-then-generate mechanism at the training stage, which means the MOSL module will predict the BIO tags for tokens in a sentence, and then the generation model will start to decode the tokens. Such a mechanism can force the text generation module to capture the boundary information of multi-word aspect/opinion terms.

When the input sentence contains multiple triplets, the aspect/opinion marker features in different positions correspond to different tagged sequences in the MOSL module, e.g., $Y_i^{ma} = \{y_{i1}^{ma}, y_{i2}^{ma}, \cdots, y_{iL}^{ma}\}$ for $\mathbf{m}_i^a$ and $Y_i^{mo} = \{y_{i1}^{mo}, y_{i2}^{mo}, \cdots, y_{iL}^{mo}\}$ for $\mathbf{m}_i^o$, where Y^{ma} and Y^{mo} are the BIO tags in sequence labeling. Hence the same marker in the generation module can share information without causing confusion since it has different pointers referring to multiple aspect/opinion terms in MOSL, which consequently benefits the decoding of the sentence containing multiple triplets. Then, we feed the marker-oriented features into a fully connected layer to predict the tags of aspect/opinion terms and get the predicted probabilities over the label set:

$$\begin{aligned}p_{ij}^{ma} &= softmax(\mathbf{W}_2\mathbf{q}_{ij}^a + \mathbf{b}_2),\\ p_{ij}^{mo} &= softmax(\mathbf{W}_2\mathbf{q}_{ij}^o + \mathbf{b}_2),\end{aligned} \quad (7)$$

The training loss for MOSL is defined as the cross-entropy loss:

$$\begin{aligned}\mathcal{L}_m^{a\to o} = &-\sum_{i=1}^{N}\sum_{j=1}^{L}\sum_{c\in\mathcal{C}}\mathbb{I}(y_{ij}^{ma}=c)\cdot log(p_{i,j|c}^{ma})\\ &-\sum_{i=1}^{N}\sum_{j=1}^{L}\sum_{c\in\mathcal{C}}\mathbb{I}(y_{ij}^{mo}=c)\cdot log(p_{i,j|c}^{mo}),\end{aligned} \quad (8)$$

where $\mathbb{I}(\cdot)$ is the indicator function, y_{ij}^{ma} and y_{ij}^{mo} are the ground truth labels, and $\mathcal{C}$ denotes the {B, I, O} label set.

Training For a better understanding of bidirectional dependency and also for less space cost, we jointly optimize two bidirectional templates for the sentence and label pair (X, T):

$$\mathcal{L} = \lambda(\mathcal{L}_g^{a\to o} + \mathcal{L}_m^{a\to o}) + (1-\lambda)(\mathcal{L}_g^{o\to a} + \mathcal{L}_m^{o\to a}), \quad (9)$$

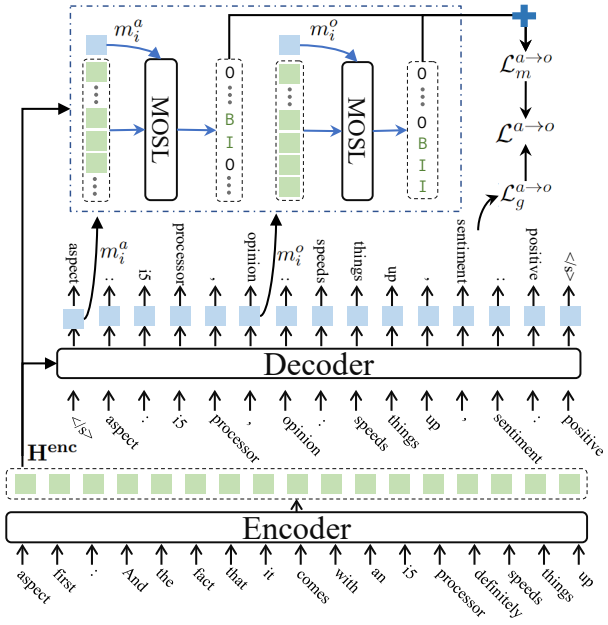

Figure 3: The process of template-guided text generation enhanced by the MOSL module.

[1]Note only aspect and opinion terms are derived from the input sentence, thus the sentiment marker is not used in MOSL.

[2]The tokenizer may split the word into several tokens. Here, we obtain the word-level features for MOSL by taking the maxpooling of the token-level features.

Dataset	Lap14					Res14					Res15					Res16				
	#S	#T	#MW	N=1	$N \geq 2$	#S	#T	#MW	N=1	$N \geq 2$	#S	#T	#MW	N=1	$N \geq 2$	#S	#T	#MW	N=1	$N \geq 2$
Train	906	1460	636	545	361	1266	2338	752	605	661	605	1013	335	338	267	857	1394	476	504	353
Dev	219	346	156	133	86	310	577	189	91	219	148	249	84	51	97	210	339	123	63	147
Test	328	543	252	184	144	492	994	337	206	286	322	485	188	210	112	326	514	170	192	134

Table 1: Statistics of the datasets. #S, #T, and N are the number of sentences, triplets, and triplets in a sentence. #MW denotes the number of triplets where at least one of aspect/opinion terms contains multiple words.

where λ is a hyper parameter to control the contributions of different templates.

3.4 Inference

Constrained Decoding (CD) During inference, we employ a constrained decoding (CD) strategy to guarantee the content and format legitimacy, which is inspired by Bao et al. (2022); Lu et al. (2021). The content legitimacy means that aspect/opinion terms should be a single word or multiple continuous words in the input sentence, and the sentiment must be either positive, neutral, or negative. The format legitimacy means that the generated sequence should meet the formatting requirements defined in the template.

Both types of legitimacy can be viewed as the constraint on the candidate vocabulary during the decoding process. Before decoding, we enumerate the candidate vocabulary for each token in the input sentence and templates. We then use the constrained decoding strategy to adjust the candidate vocabulary according to the current input token at each decoding time step. For example, when we input the start token "</s>" to the decoder, the candidate token should be "aspect"/"opinion" to guarantee the format legitimacy. When we input ":", the model needs to determine which is the first word of the aspect/opinion term, and the candidate tokens should be consistent with those in the input sentence.

Triplet De-linearization So far, we have generated two sequences Y_a and Y_o based on two input sentences X_a and X_o with the constrained decoding strategy. We then de-linearize them into two triplet sets T_a and T_o according to pre-defined templates $\psi_{a \to o}$ and $\psi_{o \to a}$. We take the intersection of T_a and T_o as the final prediction results.

4 Experiments

4.1 Datasets

Our proposed model is evaluated on four ASTE datasets released by Xu et al. (2020) which correct the missing triplets that are not explicitly annotated in the previous version (Peng et al., 2020). All datasets are based on SemEval Challenges (Pontiki et al., 2014, 2015, 2016) and consist of reviews in the laptop and restaurant domains. Table 1 shows the statistics of four benchmark datasets.

4.2 Implementation Details

As mentioned in Sec. 3.2, T5-Base (Raffel et al., 2020) is used to initialize the parameters of our model. We train our model using AdamW optimizer with an initial learning rate 3e-4 and linear learning rate decay. The number of training epoch is set to 20 for full supervised settings and 200 for low-resource and few-shot settings. When encoding the bidirectional dependency jointly, we set the batch size to 32 and λ to 0.5. The results for supervised and low-resource settings are averaged over five and ten runs with different random initialization, respectively. All experiments are conducted on an NVIDIA RTX 3090 GPU.

4.3 Baselines

To validate the effectiveness of our proposed model, we compare it with 14 state-of-art baselines. We divide the baselines into three categories. (1) **pipeline methods**: CMLA+, RINANTE+, Li-unified-R, and Peng-two-stage are proposed by Peng et al. (2020). (2) **unified non-generative methods**: JET-BERT (Xu et al., 2020), OTE-MTL (Zhang et al., 2020), GTS-BERT (Wu et al., 2020b), SPAN-ASTE (Xu et al., 2021), BMRC (Chen et al., 2021), EMC-GCN (Chen et al., 2022). (3) **generative methods**: BART-GEN (Yan et al., 2021), GAS (Zhang et al., 2021b), PARAPHRASE (Zhang et al., 2021b), SSI+SEL (Lu et al., 2022).

4.4 Main Results

Supervised settings Table 2 shows the triplet extraction performance under supervised settings. Our proposed SLGM method beats all baselines in terms of F_1 scores. Specifically, our SLGM outperforms the best text-generation method SSI+SEL by 2.48, 2.64, 4.72, and 2.54 points on four datasets, respectively. Moreover, our SLGM can exploit knowledge for triplet extraction directly from training data, contradicting to SSI+SEL's pre-training

Models	Lap14			Res14			Res15			Res16		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
CMLA+‡	30.09	36.92	33.16	39.18	47.13	42.79	34.56	39.84	37.01	41.34	41.10	41.72
RINANTE+‡	21.71	18.66	20.07	31.42	39.38	34.95	29.88	30.06	29.97	25.68	22.30	23.87
Li-unified-R‡	40.56	44.28	42.34	41.04	67.35	51.00	44.72	51.39	47.82	37.33	54.51	44.31
Peng-two-stage‡	37.38	50.38	42.87	43.24	63.66	51.46	48.07	57.51	52.32	46.96	64.24	54.21
OTE-MTL‡	49.62	41.07	44.78	62.70	57.10	59.71	55.63	42.51	47.94	60.95	53.35	56.82
JET-BERT‡	55.39	47.33	51.04	70.56	55.94	62.40	64.45	51.96	57.53	70.42	58.37	63.83
GTS-BERT‡	57.82	51.32	54.36	67.76	67.29	67.50	62.59	57.94	60.15	66.08	69.91	67.93
SPAN-ASTE‡	63.44	55.84	59.38	72.89	70.89	71.85	62.18	64.45	63.27	69.45	71.17	70.26
BMRC‡	70.55	48.98	57.82	75.61	61.77	67.99	68.51	53.40	60.02	71.20	61.08	65.75
EMC-GCN‡	61.70	56.26	58.81	71.21	72.39	71.78	61.54	62.47	61.93	65.62	71.30	68.33
BART-GEN‡	61.41	56.19	58.69	65.52	64.99	65.25	59.14	59.38	59.26	66.60	68.68	67.62
GAS†	61.65	58.19	59.87	71.08	71.67	71.37	60.01	63.67	61.78	67.76	71.67	69.66
PARAPHRASE†	62.99	58.30	60.55	70.87	70.90	70.89	60.80	64.98	62.82	70.35	74.04	72.15
SSI+SEL†	65.95	59.93	<u>62.79</u>	72.47	73.54	<u>73.00</u>	63.13	63.66	<u>63.55</u>	71.05	75.64	<u>73.26</u>
SLGM	70.54	60.74	**65.27***	78.84	72.70	**75.64***	69.75	66.85	**68.27***	75.86	75.76	**75.80***

Table 2: Results for supervised settings. The baseline results with "‡" are retrieved from Yan et al. (2021); Xu et al. (2021); Chen et al. (2022). We reproduce the generative methods with "†" by using their released code. The best and the second best F_1 scores are in **bold** and <u>underlined</u>, respectively. The $*$ marker denotes the statistically significant improvements with $p < 0.01$ over the second best results by SSI+SEL.

Dataset	Model	PLM	1-shot	5-shot	10-shot	AVG-S	1%	5%	10%	AVG-R
Lap14	SSI+SEL	UIE-base	5.27	19.06	27.77	17.37	14.98	37.02	44.51	32.17
	SLGM	T5-base	**11.95**	**31.30**	**41.53**	**28.26***	**27.14**	**47.40**	**53.72**	**42.75***
Res14	SSI+SEL	UIE-base	11.65	32.54	40.56	28.25	31.44	53.34	61.13	48.64
	SLGM	T5-base	**23.26**	**44.87**	**50.99**	**39.71***	**43.44**	**59.68**	**64.68**	**55.93***
Res15	SSI+SEL	UIE-base	10.83	28.48	38.08	25.80	17.95	39.73	48.60	35.43
	SLGM	T5-base	**22.43**	**43.44**	**51.45**	**39.11***	**30.64**	**51.35**	**57.93**	**46.64***
Res16	SSI+SEL	UIE-base	10.36	26.78	39.14	25.43	23.28	49.91	57.36	43.52
	SLGM	T5-base	**22.65**	**46.08**	**52.73**	**40.49***	**37.44**	**57.07**	**63.30**	**52.60***

Table 3: Results for low-resource settings, where AVG-S and AVG-R are the average results across 3 few-shot and 3 low-resource settings, respectively. The best F_1 scores are in **bold**. The $*$ marker denotes the statistically significant improvements with $p < 0.01$ over SSI+SEL.

method which relies on extra data like Wikipedia and Wikidata.

The generative methods like GAS which use the classic encoder-decoder architecture can outperform most non-generative methods without complicated architectures through learning label semantics. We also find that the non-generative method BMRC achieves competitive precision scores on four datasets because it also considers the bidirectional dependency. By combining the text generation and sequence labeling in training for tackling the complex extraction scenarios, our SLGM method improves the precision of GAS by more than 7 points and the recall of BMRC by more than 10 points.

Low-resource settings To validate the model's performance in the low-resource scenarios, we follow the settings in SSI+SEL (Lu et al., 2022) to conduct experiments on six different partitions of the original training sets (1/5/10-shot, 1/5/10%-ratio) and report averaged scores over random 10 runs. SSI+SEL adopts a pre-training process which can help the model capture general information from additional data. However, as shown in Table 3, our SLGM achieves much better results than SSI+SEL by a large margin on all partitions without such a pre-training process. The performance gap between our SLGM and SSI+SEL becomes more impressive under the low-resource settings than that under the supervised ones. This clearly demonstrates that our SLGM model can be quickly adapted to the low-resource scenarios with very few samples, which is an extremely good property of our model.

Mode	Lap14			Res14			Res15			Res16		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
T_a	64.12	64.46	64.28	73.79	75.61	74.69	64.19	70.43	67.16	71.15	77.51	74.19
T_o	64.20	64.27	64.24	73.29	75.45	74.35	62.70	69.36	65.86	70.43	77.43	73.76
$T_a \cap T_o$	70.54	60.74	**65.27**	78.84	72.70	**75.64**	69.75	66.85	**68.27**	75.86	75.76	**75.80**

Table 4: Impacts of bidirectional templates. T_a and T_o denote the predicted results from different decoding order.

Model	Lap14	Res14	Res15	Res16
SLGM	**65.27**	**75.64**	**68.27**	75.80
w/o $\psi_{o\to a}$	64.39	74.07	66.31	74.67
w/o $\psi_{a\to o}$	64.01	73.28	65.60	73.18
w/o MOSL	62.73	74.61	66.72	73.82
w/o CD	65.00	75.25	68.16	**75.86**

Table 5: Results for ablation study under supervised settings.

Model	1-shot	5-shot	10-shot	1%	5%	10%
SLGM w/o CD	14.85	38.78	46.30	28.19	54.38	62.28
SLGM	22.65	46.08	52.73	37.44	57.07	63.30
△	+7.80	+7.30	+6.43	+9.25	+2.69	+1.02

Table 6: Results for ablation study under low-resource settings for constrained decoding (CD) on the Res16 dataset.

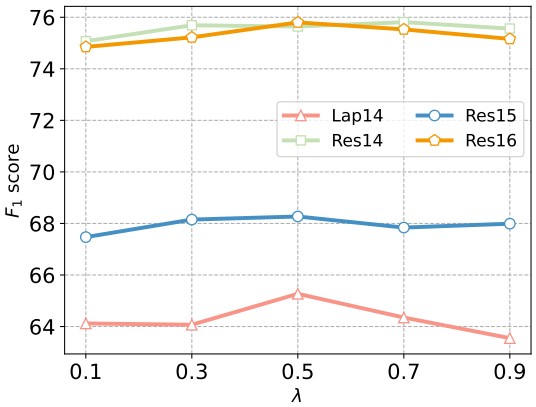

Figure 4: F_1 scores under different settings of λ.

5 Analysis

5.1 Ablation Study

To examine the impacts of three key components in our model, including marker-oriented sequence labeling (MOSL), bidirectional templates ($\psi_{a\to o}$ and $\psi_{o\to a}$), and constrained decoding (CD), we conduct the ablation study on four datasets under supervised settings, The results are shown in Table 5. We make the following notes.

Firstly, removing one of two bidirectional templates will cause a performance drop, and $\psi_{a\to o}$ contributes more to the model than $\psi_{o\to a}$.

Secondly, the extraction performance decreases dramatically after removing MOSL. This clearly proves the effectiveness of MOSL module. We will make more exploration about the impacts of MOSL in Sec. 5.3.

Thirdly, "w/o CD" denotes that we directly take the whole vocabulary instead of taking the format and content constraints into account. We find that the performance slightly degrades on Lap14, Res14, and Res15, but increases on Res16. The reason might be that limiting the size of candidate vocabulary leads the model to generate some wrong but legal triplets. However, the large amount of training data under the supervised settings allows the model to adaptively fit to the target text.

To confirm this hypothesis, we further investigate the impacts of CD under the low-resource settings on the Res16 dataset [3]. The results are shown in Table 6. We can see that as the number of training samples decreases, the performance gain from CD becomes more significant. This infers that the CD strategy plays a more important role in data scarcity scenario.

5.2 Impacts of Bidirectional Templates

We model the mutual dependency between aspect and opinion terms using the bidirectional templates. Our purpose is to avoid generating false paired aspect-opinion triplets. We investigate the impacts of bidirectional templates and show the results in Table 4. Besides, we also plot the performance under different settings of λ to further validate the importance of bidirectional dependency as shown in Fig. 4.

It can be seen that the unidirectional decoding order T_a/T_o gets better recall scores but generates many false triplets, and thus has low precision. By capturing the mutual dependency and taking the intersection of T_a and T_o, our model can effectively filter false paired triplets and significantly enhance

[3]We have similar observations on other datasets. We omit those results for clarity.

Mode	Model	Lap14	Res14	Res15	Res16
Single Word	SLGM w/o MOSL	71.00	80.10	71.70	78.58
	SLGM	72.22	80.88	73.35	80.44
	△	+1.22	+0.78	+1.65	+1.86
Multi Word	SLGM w/o MOSL	52.34	62.93	58.69	63.85
	SLGM	56.63	64.56	60.22	66.19
	△	+4.29	+1.63	+1.53	+2.34
Single Triplet	SLGM w/o MOSL	66.42	74.25	67.07	70.55
	SLGM	68.16	74.72	67.72	72.64
	△	+1.74	+0.47	+0.65	+2.09
Multi Triplet	SLGM w/o MOSL	60.45	74.72	66.41	76.00
	SLGM	63.55	75.91	68.75	77.93
	△	+3.10	+1.19	+2.34	+1.93

Table 7: Impacts of the MOSL module with different evaluation modes.

the precision and F_1 scores. Moreover, when λ is biased towards $\psi_{a\to o}$ or $\psi_{o\to a}$, the performance tends to decrease. Meanwhile, when λ is set to 0.5, the model achieves optimal results on most of the datasets. This further confirms that the bidirectional dependency is of the same importance.

5.3 Impacts of Marker-Oriented Sequence Labeling (MOSL)

Table 1 shows that multi-word triplets account for roughly one-third of all triplets while about half of the sentences are multi-triplet ones. Our MOSL module allows the model to learn the prompt information of aspects and opinions based on our tag-then-generate mechanism during training, which improves the model's ability of handling complex structures. We verify the effects of MOSL in this section [4].

Table 7 shows the performance with two different evaluation modes, where "Single-Word" denotes both aspect and opinion terms in a triplet are single-word spans, and "Multi-Word" denotes that at least one of the aspect or opinion terms in a triplet is a multi-word span. We find that the model obtains more significant improvements for multi-word triplets than that for single-word triplets after adding the MOSL module. It shows that the model can learn the boundary information of aspect/opinion terms and generate the complete terms with the guidance of MOSL.

Table 7 also presents the results for "Single-" or "Multi-" triplets in a sentence, where the MOSL module makes the similar contributions. As can be seen, the model with MOSL gains more improvements when the review contains multiple triplets. In addition, we attempt to mix the test sets of datasets Res14, Res15, and Res16 to evaluate the performance of the model under multi-triplet setting [5]. The ratio of the averaged improvement of the multi-triple to the single-triple setting on three single dataset is 1.77 while it increases up to 3.15 on the mixed dataset. This is because all aspect/opinion features in MOSL point to the same marker "aspect/opinion". This allows the marker to share knowledge across different aspect/opinion features, thus the text generation module holds the clue from the shared marker about the subsequent aspect/opinion term when generating the prior ones.

[4]Note that the sentences with multi-word triplets and the multi-triplet sentences overlap in many cases. Hence the impacts of MOSL may not clearly present as expected on some datasets like Res15 or Res16.

Model	Parameter	Inference Time
GAS	222.9M	24.37S†
SLGM	225.2M	24.79S
w/o CD	225.2M	11.39S
w/o CD & MOSL	222.9M	11.02S
w/o CD & MOSL & $\psi_{o\to a}$	222.9M	5.50S

Table 8: Complexity analysis on Lap14 dataset. The results marked with † are reproduced based on the released code.

5.4 Analysis on Computational Cost

To demonstrate that our model does not bring too much computational cost, we compare it with GAS in terms of the number of parameters and inference time as shown in Table 8. We also analyze the costs of the key components in our model to show their impact on complexity. Firstly, the MOSL module adds only about 2.3M parameters compared with GAS. Secondly, we find that the constrained decoding algorithm increases the inference time as our implementation of constrained decoding algorithm requires determining the candidate vocabulary according to the current input token at each decoding time step, which undermines the parallelism of the generation model during inference. Moreover, bidirectional templates require the model to generate target sequences based on two different decoding orders which also increases inference time to some extent. However, SLGM does not show significant differences from GAS in terms of model parameters and inference time because GAS needs to take a prediction normalization strategy to refine the

[5]Here we still take the training set of Res15 for training.

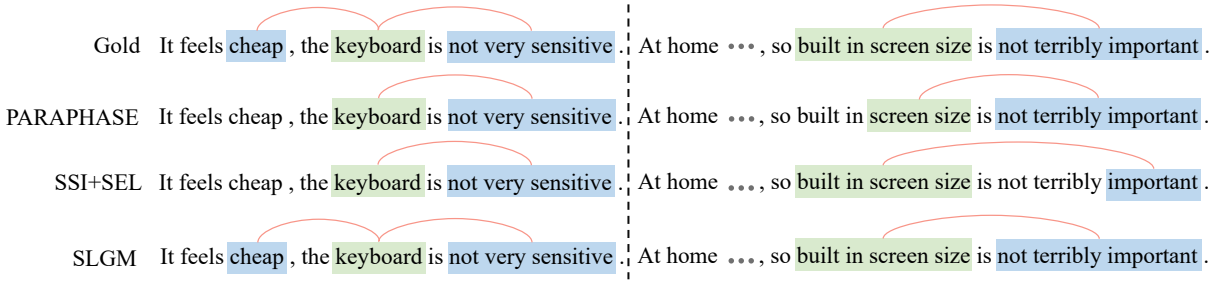

Figure 5: Case Study. The aspect and opinion terms are highlighted in green and blue, respectively. The orange line denotes the aspect term matches the opinion term and the model correctly predicts the sentiment polarity.

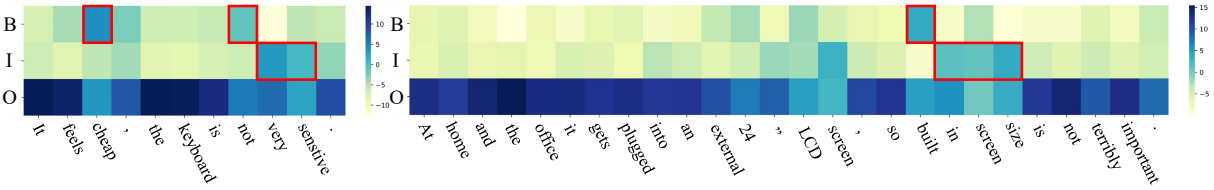

Figure 6: Visualization of the sequence labeling (BIO) probability output by MOSL. Left: the probability for the first opinion marker in the first review; Right: the probability for the first aspect marker in the second review.

prediction results.

5.5 Case Study

We conduct a case study on two reviews to compare typical generative methods, including PARAPHRASE (Zhang et al., 2021a), SSI+SEL (Lu et al., 2022), and our method. The results are as shown in Fig. 5.

For the first review (the left one in Fig. 5), SSI+SEL and PARAPHRASE cannot recognize the opinion term "cheap", whereas "not very sensitive" is recognized by all methods. In contrast, our SLGM can identify both terms. To have a close look, we further visualize the BIO probabilities output by MOSL in Fig. 6. As we can see in the left part of Fig. 6, the opinion marker in MOSL focuses on two opinion terms simultaneously when the generation module generates the first triplet, which helps the model know that there are two related opinion terms for the aspect term "keyboard".

For the second review (the right one in Fig. 5), both SSI+SEL and PARAPHRASE find the approximate locations of the aspect and opinion terms, but neither of them gets correct pairs due to incomplete decoding. The reason is that these two methods lack the corresponding prompt information for boundary identification. Meanwhile, as can be seen from the right part of Fig. 6, the aspect marker in MOSL focuses on the complete aspect term, which contains the boundary information that can help our generation module to decode the complete aspect term.

6 Conclusion

In this paper, we exploit the power of text generation and sequence labeling for ASTE. We propose two bidirectional templates to reflect the mutual aspect-opinion dependency for filtering false paired triplets. We also present a marker-oriented sequence labeling module to help the text generation module tackle complex structures in the subsequent decoding process. Experiment results show that our framework consistently outperforms all generative and non-generative baselines under both the full supervised and low-resource settings.

Limitations

Although our proposed method achieves the state-of-art performance, it still has a few limitations. Firstly, we only consider the dependency between aspect and opinion in the target text yet ignoring the order influence in the input text, which may bring more improvements. Secondly, there are three label types for ASTE, including aspect, opinion, and sentiment. Currently, we only utilize the aspect and opinion markers in the marker-oriented sequence labeling module. We believe that the specific design for the sentiment marker can further improve the performance, which can be a future direction.

Acknowledgments

This work was supported by a grant from the National Natural Science Foundation of China (NSFC) project (No. 62276193).

References

Xiaoyi Bao, Wang Zhongqing, Xiaotong Jiang, Rong Xiao, and Shoushan Li. 2022. Aspect-based Sentiment Analysis with Opinion Tree Generation. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, pages 4044–4050, Vienna, Austria. International Joint Conferences on Artificial Intelligence Organization.

Hao Chen, Zepeng Zhai, Fangxiang Feng, Ruifan Li, and Xiaojie Wang. 2022. Enhanced Multi-Channel Graph Convolutional Network for Aspect Sentiment Triplet Extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2974–2985, Dublin, Ireland. Association for Computational Linguistics.

Shaowei Chen, Yu Wang, Jie Liu, and Yuelin Wang. 2021. Bidirectional machine reading comprehension for aspect sentiment triplet extraction. In *Proceedings Of The AAAI Conference On Artificial Intelligence*, volume 35, pages 12666–12674.

Zhuang Chen and Tieyun Qian. 2020a. Enhancing Aspect Term Extraction with Soft Prototypes. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2107–2117, Online. Association for Computational Linguistics.

Zhuang Chen and Tieyun Qian. 2020b. Relation-Aware Collaborative Learning for Unified Aspect-Based Sentiment Analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3685–3694, Online. Association for Computational Linguistics.

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, Florence, Italy. Association for Computational Linguistics.

Lei Gao, Yulong Wang, Tongcun Liu, Jingyu Wang, Lei Zhang, and Jianxin Liao. 2021. Question-Driven Span Labeling Model for Aspect–Opinion Pair Extraction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(14):12875–12883. Number: 14.

Minghao Hu, Yuxing Peng, Zhen Huang, Dongsheng Li, and Yiwei Lv. 2019. Open-Domain Targeted Sentiment Analysis via Span-Based Extraction and Classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 537–546, Florence, Italy. Association for Computational Linguistics.

Zhengyan Li, Yicheng Zou, Chong Zhang, Qi Zhang, and Zhongyu Wei. 2021. Learning Implicit Sentiment in Aspect-based Sentiment Analysis with Supervised Contrastive Pre-Training. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 246–256, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yaojie Lu, Hongyu Lin, Jin Xu, Xianpei Han, Jialong Tang, Annan Li, Le Sun, Meng Liao, and Shaoyi Chen. 2021. Text2Event: Controllable Sequence-to-Structure Generation for End-to-end Event Extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2795–2806, Online. Association for Computational Linguistics.

Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2022. Unified structure generation for universal information extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5755–5772, Dublin, Ireland. Association for Computational Linguistics.

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 AAAI Conference on Artificial Intelligence*, volume 35, pages 13543–13551.

Shinhyeok Oh, Dongyub Lee, Taesun Whang, IlNam Park, Seo Gaeun, EungGyun Kim, and Harksoo Kim. 2021. Deep Context- and Relation-Aware Learning for Aspect-based Sentiment Analysis. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 495–503, Online. Association for Computational Linguistics.

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 AAAI Conference on Artificial Intelligence*, volume 34, pages 8600–8607.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, Véronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeniy Kotelnikov, Nuria Bel, Salud María Jiménez-Zafra, and Gülşen Eryiğit. 2016. SemEval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 19–30, San Diego, California. Association for Computational Linguistics.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. SemEval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 486–495, Denver, Colorado. Association for Computational Linguistics.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Meixi Wu, Wenya Wang, and Sinno Jialin Pan. 2020a. Deep Weighted MaxSAT for Aspect-based Opinion Extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5618–5628, Online. Association for Computational Linguistics.

Shengqiong Wu, Hao Fei, Yafeng Ren, Donghong Ji, and Jingye Li. 2021. Learn from Syntax: Improving Pair-wise Aspect and Opinion Terms Extraction with Rich Syntactic Knowledge. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pages 3957–3963, Montreal, Canada. International Joint Conferences on Artificial Intelligence Organization.

Zhen Wu, Chengcan Ying, Fei Zhao, Zhifang Fan, Xinyu Dai, and Rui Xia. 2020b. Grid Tagging Scheme for Aspect-oriented Fine-grained Opinion Extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2576–2585, Online. Association for Computational Linguistics.

Hu Xu, Bing Liu, Lei 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 (Volume 2: Short Papers)*, pages 592–598, Melbourne, Australia. Association for Computational Linguistics.

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 and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4755–4766, Online. Association for Computational Linguistics.

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 (EMNLP)*, pages 2339–2349, Online. Association for Computational Linguistics.

Hang Yan, Junqi Dai, Tuo Ji, Xipeng Qiu, and Zheng Zhang. 2021. A Unified Generative Framework for Aspect-based Sentiment Analysis. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2416–2429, Online. Association for Computational Linguistics.

Chen Zhang, Qiuchi Li, Dawei Song, and Benyou Wang. 2020. A Multi-task Learning Framework for Opinion Triplet Extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 819–828, Online. Association for Computational Linguistics.

Mi Zhang and Tieyun Qian. 2020. Convolution over Hierarchical Syntactic and Lexical Graphs for Aspect Level Sentiment Analysis. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3540–3549, Online. Association for Computational Linguistics.

Wenxuan Zhang, Yang Deng, Xin Li, Yifei Yuan, Lidong Bing, and Wai Lam. 2021a. Aspect sentiment quad prediction as paraphrase generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9209–9219, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. 2021b. Towards Generative Aspect-Based Sentiment Analysis. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 504–510, Online. Association for Computational Linguistics.

He Zhao, Longtao Huang, Rong Zhang, Quan Lu, and Hui Xue. 2020. SpanMlt: A Span-based Multi-Task Learning Framework for Pair-wise Aspect and Opinion Terms Extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3239–3248, Online. Association for Computational Linguistics.

Yuxiang Zhou, Lejian Liao, Yang Gao, Zhanming Jie, and Wei Lu. 2021. To be closer: Learning to link up aspects with opinions. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3899–3909, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Limitations

☐ A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 4.1

☑ B1. Did you cite the creators of artifacts you used?
Section 4.1

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section 4.1

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 4.1

C ☑ Did you run computational experiments?

Section 4, Section 5.1-5.4

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 5.4

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 4.2

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 4.2

☐ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Not applicable. Left blank.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.